

UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR



**Grado en Ingeniería de Tecnologías y Servicios de la
Telecomunicación**

TRABAJO FIN DE GRADO

**Análisis estadístico de ciberataques mediante distribuciones
alfaestables**

Eduardo Revuelta Santiago

Tutor: Luis de Pedro Sánchez

Ponente (si procede): Jorge López de Vergara

Julio 2020

Análisis estadístico de ciberataques mediante distribuciones alfaestables

AUTOR: Eduardo Revuelta Santiago

TUTOR: Luis de Pedro Sánchez

**Computación y redes de altas prestaciones
Dpto. Tecnología Electrónica y de las Comunicaciones
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Julio de 2020**

Resumen (castellano)

El avance de la tecnología durante la última década nos ha traído nuevas máquinas sofisticadas que hace pocos años, jamás abríamos imaginado que pudiesen existir. Su desarrollo ha desencadenado la eterna carrera de la ciberseguridad: el gato y el ratón corriendo en una carrera sin fin por ver quien adelanta al otro. Establecer protocolos y medidas para que estas máquinas puedan funcionar de forma eficiente y segura ha sido una prioridad desde el principio

Esto no ha impedido que los ataques cibernéticos se hayan erradicado, sino todo lo contrario: el aumento de las medidas de seguridad ha servido de motivación para la creación de nuevas técnicas para saltar dichas barreras y dejar nuestros dispositivos vulnerables.

La ciberseguridad afronta nuevos retos todos los días, y de ahí que deban surgir nuevas técnicas, como algunos modelos estadísticos que intentan describir y parametrizar el tráfico de red para poder analizarlo y conocer mejor sus características, hasta el punto de poder detectar anomalías.

Precisamente, este trabajo se centra intentar conocer mejor el tráfico de red analizándolo mediante la distribución estadística alfaestable, para así ser capaces de detectar un ataque informático cuando a simple vista parece invisible.

En este trabajo se continúa analizando una trama de tráfico real, tomando de punto de partida el trabajo y las conclusiones obtenidas por los autores previos, para así realizar una labor de investigación sobre el citado tráfico de red y ver si hay diferencias entre los parámetros estadísticos del tráfico normal y el tráfico de ataque.

Por tanto, se trabajará con dos series de datos, un ataque informático sintético y el tráfico real de red. Mediante métodos de agrupación y de clasificación se pretenderá encontrar alguna forma de aislar el tráfico de ataque frente al normal para que pueda ser detectado.

Palabras clave (castellano)

Ciberataque, MATLAB, denegación de servicio (DoS), distribución alfaestable, *K-Means*, clasificación, algoritmo, *clustering*.

Abstract

The advances made in technology during the last decade have brought to us new sophisticated machines that, few years back, we could not even think of existing. Its development has unchained the eternal race of cybersecurity: the mice and the cat running the never-ending race to see which one of them overtakes the other. Establishing protocols and measures to make these machines work in a safe and efficient way has been a priority since the beginning of it.

However, this has not prevented cyberattacks from being eradicated, but right the opposite: the increasing amount of security measures has led to a rise of motivation on the creation of new techniques to surpass these barriers and make our devices vulnerable.

Cybersecurity faces new challenges every day and it is the reason why new techniques must arise, such as some statistic models that can be used to describe and parameterize network traffic in order to analyze it and learn about its characteristics, to the extent where anomalies can be detected.

Precisely, this piece of work will focus on getting to know better network traffic, analyzing it through the alpha-stable distribution, to see if we are able to detect an attack that looks invisible to the eye.

In this piece of work, we will continue analyzing a real frame of network traffic, taking as the starting point all the work done and the conclusions made by the previous authors, in order to perform a work of investigation within said frame to see whether or not there are differences between the statistic parameters of the normal traffic and the attack traffic.

Therefore, two series of data will be analyzed, one containing the real network traffic and the other one containing the attack. Through clustering and classification methods, we will try to find some sort of way to isolate attack traffic from the normal traffic to be detected.

Keywords

Cyberattack, MATLAB, Denial of Service (DoS), Alpha-stable distribution, K-Means, classification, algorithm, clustering.

Agradecimientos

Me gustaría agradecer a mi tutor Luis de Pedro Sánchez y a mi ponente Jorge López de Vergara todo el apoyo mostrado, sin cuya dedicación este trabajo no podría haberse llevado a cabo.

Por otro lado, me gustaría agradecer a todos mis amigos y familiares, que me motivan cada día para continuar esforzándome, en especial a mi madre y a mi padre.

INDICE DE CONTENIDOS

1 Introducción.....	1
1.1 Motivación.....	1
1.2 Objetivos.....	1
1.3 Fases de realización.....	2
1.4 Organización de la memoria.....	3
2 Estado del arte	4
2.1 Introducción.....	4
2.2 Ciberataques	4
2.3 Distribución α -estable.....	5
2.4 Algoritmos de clasificación y agrupación	6
2.5 Coeficiente de silueta.....	8
2.6 Conclusiones.....	10
3 Diseño.....	11
3.1 Introducción.....	11
3.2 MATLAB	11
3.3 Etiquetas de clasificación	12
3.4 Algoritmo de CL.....	13
3.5 Coeficiente de silueta.....	14
3.6 Conclusiones.....	15
4 Desarrollo	16
4.1 Introducción.....	16
4.2 Series temporales	16
4.3 Representaciones de los parámetros	20
4.4 Algoritmo de clasificación	23
4.5 Algoritmo de clasificación mediante centroides	27
4.6 Coeficiente de silueta.....	28
4.7 Datos desbalanceados	29
4.8 Algoritmo de centroides con tráfico particionado	30
4.8.1 Número de centroides.....	30
4.8.2 Tráfico particionado	31
4.8.3 Metodología.....	31
4.9 Conclusiones.....	33
5 Pruebas y resultados	34
5.1 Introducción.....	34
5.2 Metodología.....	34
5.3 Resultados.....	35
5.4 Conclusiones.....	37
6 Conclusiones y trabajo futuro.....	38
6.1 Conclusiones.....	38
6.2 Trabajo futuro	38
Referencias	39
Glosario	40
Anexos	I
A Gráficos adicionales.....	I
B Repositorio del código generado	- 1 -

INDICE DE FIGURAS

FIGURA 1: DIAGRAMA DE GANTT.	3
FIGURA 2: “CHULETA DE SELECCIÓN DE ALGORITMO” [5]	7
FIGURA 3: GRÁFICO DE SILUETA. [9].....	9
FIGURA 4: COMPARATIVA DE LENGUAJES.....	11
FIGURA 5: TIPOS DE TRÁFICO.	13
FIGURA 6: DIAGRAMA DE BLOQUES DEL DESARROLLO.	16
FIGURA 7: SERIE TEMPORAL DE ALFA.....	17
FIGURA 8: SERIE TEMPORAL DE BETA.....	17
FIGURA 9: SERIE TEMPORAL DE DELTA.	18
FIGURA 10: SERIE TEMPORAL DE GAMMA.	18
FIGURA 11: SERIE TEMPORAL DE DELTA EN LA ZONA DEL ATAQUE.....	19
FIGURA 12: REPRESENTACIÓN DE BETA FRENTE A DELTA.	20
FIGURA 13: REPRESENTACIÓN DE GAMMA FRENTE A DELTA.....	21
FIGURA 14: REPRESENTACIÓN DE ALFA FRENTE A DELTA.	21
FIGURA 15: REPRESENTACIÓN DE ALFA FRENTE A GAMMA.....	22
FIGURA 16: CLASIFICACIÓN ALFA FRENTE A DELTA	24
FIGURA 17: CLASIFICACIÓN DE GAMMA FRENTE A DELTA.	24
FIGURA 18: CLASIFICACIÓN DE ALFA FRENTE A GAMMA.....	25
FIGURA 19: MATRIZ DE CONFUSIÓN ALFA-DELTA.	26
FIGURA 20: MATRIZ DE CONFUSIÓN ALFA-GAMMA-DELTA.....	26
FIGURA 21: GRÁFICOS DE SILUETA CON NÚMERO ÓPTIMO DE CENTROIDES.	28
FIGURA 22: MATRIZ DE CONFUSIÓN CON DATOS DESBALANCEADOS.	29
FIGURA 23: MEJOR CLASIFICACIÓN: SEGMENTO N°4 CON ATAQUE SITUADO EN 30%	32
FIGURA 24: PEOR CLASIFICACIÓN: SEGMENTO N°9 CON ATAQUE SITUADO EN 60%.....	32

FIGURA 25: CENTROIDES DE ATAQUE SEGÚN LA POSICIÓN DEL ATAQUE.	35
FIGURA 26: CENTROIDES DEL SEGMENTO N°9.....	36
FIGURA 27: REPRESENTACIÓN 3D DE LA COMPROBACIÓN	37
FIGURA 28: REPRESENTACIÓN DE ALFA FRENTE A GAMMA DE LA COMPROBACIÓN	37

1 Introducción

En este apartado se introduce los hechos que han motivado la realización de este TFG al igual que la estructura que sigue, las fases que se han llevado a cabo y los objetivos que se han buscado.

1.1 Motivación

Al desarrollo del sector informático y tecnológico en los últimos le ha acompañado un creciente aumento de ciberataques, que cada vez aumentan más su alcance y su capacidad de pasar desapercibidos, afectando así a nuevos dispositivos que salen en el mercado y poniendo en peligro la infraestructura computacional de la red, los usuarios y su información. Se deben garantizar ciertos derechos como el derecho a la privacidad y el derecho a la integridad de la información para que el funcionamiento de la red sea correcto.

Esto ha desencadenado la necesidad de estudiar el tráfico de red, atendiendo a los distintos tipos de ataque que se dan. Entre ellos se encuentra el ataque de denegación de servicio (en inglés *Denial of Service*, DoS) que resulta ser uno de los más utilizados para atacar dichas infraestructuras computacionales y comprometer el estado de la red. Sus consecuencias pueden ser devastadoras, sobre todo para empresas que necesitan satisfacer el servicio a un cliente, y son relativamente sencillos de ejecutar. Esta es la razón por la que cada vez más empresas opten en invertir en un departamento de ciberseguridad.

Anticiparnos a los ataques de denegación de servicio (y a cualquier ataque) es de vital importancia. Para poder prevenirlos es necesario primero detectarlos. Mediante los parámetros de la distribución alfaestable conseguimos distinguir los dos tipos de tráfico: el normal y el ataque, pero para detectarlos correctamente es necesario encontrar un patrón entre ambos tipos de tráfico que nos permita diferenciarlos.

En este trabajo se analizan las diferencias entre ambos tipos de tráfico para así desarrollar un algoritmo que nos permita reconocer y detectar el ataque DoS en un entorno real, y realizar pruebas con dicho algoritmo para analizar sus resultados.

1.2 Objetivos

El objetivo de este trabajo es volver analizar los resultados obtenido en el TFG previo para así poder profundizar en la creación de un algoritmo que nos ayude a clasificar de forma efectiva un ataque DoS cuando ocurra en una trama de tráfico real.

El proceso llevado a cabo se divide en las siguientes fases:

- Parametrizar el tráfico
- Agrupar el tráfico según su tipo
- Clasificar del tráfico según su tipo
- Distinguir tipos de tráfico

1.3 Fases de realización

En este apartado, se dará una breve descripción de cada una de las fases del trabajo introducidas en el apartado anterior:

- **Series temporales:** Realización de un estudio de las representaciones temporales de cada uno de los parámetros alfaestables gracias al material provisto por el TFG de partida.
- **Representación de parámetros:** Representaciones bidimensionales de los parámetros alfaestables en las cuales se distinguen los distintos tipos de tráfico.
- **Algoritmo de clasificación:** Tras realizar las representaciones de los parámetros se procede a seleccionar cuales de ellas serán más interesantes de analizar. Posteriormente, se creará un clasificador de tráfico que permita distinguirlos y a partir del cual se pueda crear un predictor que sea eficaz a la hora de identificar si se está ante un ataque.
- **Algoritmo de clasificación mediante centroides:** El objetivo de este apartado es desarrollar un algoritmo que calcule los centroides del tráfico completo y sea capaz de clasificarlos de forma efectiva.
- **Coeficiente de silueta:** Se analizan los coeficientes de silueta de los centroides para optimizar el algoritmo.
- **Datos desbalanceados:** Se explica por qué surge el problema de trabajar con datos desbalanceados y cuáles son sus implicaciones.
- **Algoritmo de centroides con tráfico particionado:** Se implementa un conjunto de modificaciones a al algoritmo para solucionar el problema que supone trabajar con datos desbalanceados.
- **Análisis de los resultados:** Tras realizar las ejecuciones y recopilar los resultados, se procede a analizarlos y sacar conclusiones.

Se dispone de un diagrama de Gantt el cual muestra el tiempo dedicado durante el curso a cada una de las fases anteriormente explicadas:

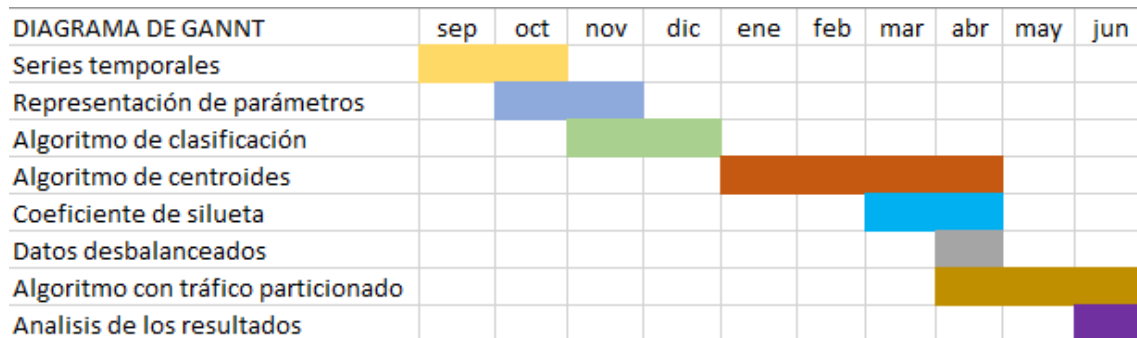


Figura 1: Diagrama de Gantt.

1.4 Organización de la memoria

La estructura de la memoria se organiza de la siguiente manera:

- **Capítulo 1: Introducción:** Se expone una breve introducción del TFG, que sirve al lector para entender mejor el trabajo desarrollado a continuación.
- **Capítulo 2: Estado del Arte:** Explicación de los conceptos clave necesarios para entender las ideas que se tratan en el trabajo.
- **Capítulo 3: Diseño:** Descripción de las herramientas utilizadas para el diseño de este trabajo.
- **Capítulo 4: Desarrollo:** Explicación detallada de todas las fases del trabajo, explicando por que se ha llevado a cabo cada una de ellas.
- **Capítulo 5: Pruebas y resultados:** Análisis de todos los resultados obtenidos en las distintas pruebas realizadas.
- **Capítulo 6: Conclusiones y trabajo futuro:** Se exponen las conclusiones obtenidas tras los resultados, a partir de las cuales se proponen posibles trabajos futuros.

2 Estado del arte

2.1 Introducción

En este capítulo se expondrán todos los conceptos necesarios para poder entender mejor el trabajo que se ha desarrollado en los siguientes capítulos. Los conceptos son los siguientes:

- Ciberataques
- Distribución α -estable
- Algoritmos de clasificación y agrupación
- Coeficiente de silueta

A continuación, se procede a explicarlos en detalle.

2.2 Ciberataques

Los ciberataques [1] o ataques informáticos son aquellos que son realizados a través de la red, cuyo objetivo es dañar un sistema informático de forma no autorizada y con un fin malicioso. Los ataques informáticos pueden ser de varios tipos según su objetivo. Estos pueden ser: tomar el control o desestabilizar un sistema informático para corromper archivos, datos privados o algoritmos de dicho sistema.

Algunos de ellos son el ataque de denegación de servicio [2] y su versión distribuida, los cuales se explicarán a continuación debido su estrecha relación con el trabajo realizado.

Ataque de denegación de servicio (DoS): Este ataque informático tiene como propósito inutilizar un sistema informático de forma que su servicio de red quede bloqueado. Los servidores de la red pueden atender a un número limitado de peticiones y si este se sobrepasa, el sistema puede sufrir ralentizaciones o incluso bloqueos y desconexiones de la red. Los ataques DoS se caracterizan por generar un número enorme de peticiones desde una sola máquina que tienen como objetivo inundar la máquina o el sistema víctima consumiendo todos sus recursos hasta que empieza a rechazar peticiones. La práctica de este ataque ha crecido con el paso del tiempo, debido a la relativa facilidad que posee de ejecutarse.

En el ataque distribuido de denegación de servicio (en inglés *Distributed Denial of Service*, DDoS) se emiten las peticiones desde múltiples máquinas. Los paquetes maliciosos se envían al mismo tiempo y a la misma víctima. Sus consecuencias son que el ataque sea más difícil de rastrear y de bloquear, ya que dicho ataque proviene de distintas direcciones IP, y que tenga una mayor capacidad de derribar el servicio ya que dispone de un gran número de máquinas, por lo que la cantidad de peticiones suele ser muy superior a la de un ataque DoS.

Las máquinas que participan en el ataque se las denomina “Bots” o “Zombis”, ya que son infectados con un malware cuyo objetivo es que puedan ser controlados de forma remota por el hacker o ciberdelincuente.

En la actualidad, se investiga para desarrollar técnicas más avanzadas con las que detectar y prevenir este tipo de ataques.

2.3 Distribución α -estable

Las distribuciones estables o α -estable [3] son aquellas que se caracterizan por ser una combinación lineal de dos o más réplicas independientes de una muestra aleatoria que posee la misma distribución de probabilidad. Esta definición se basa en que una suma de dos variables aleatorias siempre es una variable aleatoria. Por lo tanto, si se suman dos variables aleatorias alfa-estables, su resultado también será una variable aleatoria alfa-estable.

Una distribución es α -estable si se cumple la siguiente propiedad:

Dadas dos copias independientes de una variable aleatoria X , llamadas X_1 e X_2 , y se cumple que:

$$aX_1 + bX_2 = cX + d$$

Siendo a , b y c tres constantes cualesquiera tal que $a > 0$, $b > 0$ y $c > 0$ y d otra constante real. Entonces, si ambas dos variables aleatorias ($[aX_1 + bX_2]$ y $[cX + d]$) tienen la misma distribución, se puede decir que X es estable. Y si esto se sigue cumpliendo para un valor de $d = 0$, entonces podremos decir que la distribución es estrictamente estable.

Podemos escribir la distribución de probabilidad a partir de su función característica, dado que estas distribuciones no admiten una expresión de su densidad de probabilidad:

$$f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \varphi(t) e^{-ixt} dt \quad (1)$$

Si $\varphi(t)$ es la función característica de la distribución α -estable, entonces podremos definirla mediante la siguiente expresión:

$$\varphi(t; \alpha, \beta, c, \mu) = \exp[it\mu - |ct|^\alpha (1 - i\beta \operatorname{sgn}(t)\Phi)] \quad (2)$$

Donde $\operatorname{sgn}(t)$ es la función signo de t y Φ se puede expresar como:

$$\Phi = \tan(\pi\alpha/2) \quad (3)$$

Excepto para $\alpha = 1$, que, debido a la indeterminación resultante de la expresión anterior, se expresa de la siguiente forma:

$$\Phi = -\frac{2}{\pi} \log |t|. \quad (4)$$

Los parámetros estadísticos que describen las distribuciones alfaestables son alfa, beta, gamma (que también es conocido como “c”) y delta (también conocido como “μ”). A continuación, se explica lo que representa cada parámetro alfaestable dentro de la distribución:

- **Alfa (α):** Es el parámetro de estabilidad y es el más significativo de los cuatro. Especifica el comportamiento asintótico de dicha distribución. En las distribuciones α -estables, alfa se comprende entre 0 y 2 ($0 < \alpha \leq 2$).
- **Beta (β):** Es el parámetro de asimetría. Define la simetría de la función y en las distribuciones estables se comprende entre los valores de -1 y 1 ($-1 < \beta < 1$).
- **Gamma(γ):** Es el parámetro de escala. Mide el ancho típico de la distribución. Puede tomar cualquier valor real positivo ($0 < \gamma < \infty$).
- **Delta(δ):** Es el parámetro de localización. Pertenecce al conjunto de los números reales, pudiendo tomar valores negativos a diferencia de gamma ($-\infty < \delta < \infty$).

El motivo por el cual las condiciones explicadas anteriormente sean las que determinen si una distribución es alfaestable o no, se debe a que la función característica de la suma de dos variables aleatorias es el producto de ambas funciones características. Esto origina que al sumar dos variables aleatorias alfaestables, se obtenga otra variable aleatoria alfaestable con los mismos parámetros de alfa y beta, que se traduce en misma estabilidad y simetría; pero distintos valores de gamma y delta, que implica un cambio en la localización y en la escala.

Dentro de la función alfaestable se dan algunos casos particulares. El valor de alfa igual a 1 corresponde con la distribución de Cauchy. Las distribuciones que se comprenden entre los valores de alfa $0 < \alpha < 2$ se denominan “distribuciones de Pareto-Lévy”.

2.4 Algoritmos de clasificación y agrupación

Un algoritmo [4] es un conjunto de reglas, instrucciones y operaciones que consigue dar con la solución a un problema de forma sistemática y ordenada. Es decir, un algoritmo consigue convertir los datos de un problema (entrada) en una solución (salida) mediante un conjunto finito de pasos.

En general, no hay una definición formal precisa de algoritmo, pero todas las definiciones tienen en común las siguientes tres propiedades sobre los algoritmos:

- **Tiempo secuencial:** Los algoritmos funcionan mediante pasos, describiendo así una secuencia de estados computacionales por cada entrada
- **Estado abstracto:** Los estados computacionales de los algoritmos se pueden determinar por estructuras de primer orden.
- **Exploración acotada:** Dentro del algoritmo se dan un número concreto de pasos entre cada uno de los estados computacionales.

Por tanto, dentro de la definición de algoritmo puede entrar cualquier cosa que funcione paso a paso, cuyos pasos queden descritos de forma clara y precisa, y que cada paso posea un número limitado de datos que pueda manejar.

El aprendizaje automático (en inglés *Machine Learning*, ML) [5] se desarrolla por medio de algoritmos y su objetivo es producir un modelo mediante el cual podamos realizar una tarea. El modelo se entrena con numerosas cantidades de muestras para que pueda llegar a realizar predicciones. Dentro del ML encontramos tres categorías diferentes:

- **Aprendizaje Supervisado:** En el aprendizaje supervisado, los datos se tratan como etiquetas y los algoritmos tratan de asignar en las salidas la etiqueta correcta a cada uno de los datos de entrada. Para ello, el algoritmo se entrena con una base de datos para que aprenda a asignar las etiquetas correctamente. Este tipo de aprendizaje suele darse en problemas de regresión y de clasificación.
- **Aprendizaje no supervisado:** En el aprendizaje no supervisado, no se entrena al algoritmo en cuestión porque no se dispone de datos de entrenamiento, los únicos datos que maneja el algoritmo son las muestras de entrada. Este tipo de aprendizaje se da en algoritmos de agrupación (en inglés *Clustering*, CL) entre otros.

La siguiente figura muestra los distintos tipos de algoritmos dentro de ML, así como sus características generales:

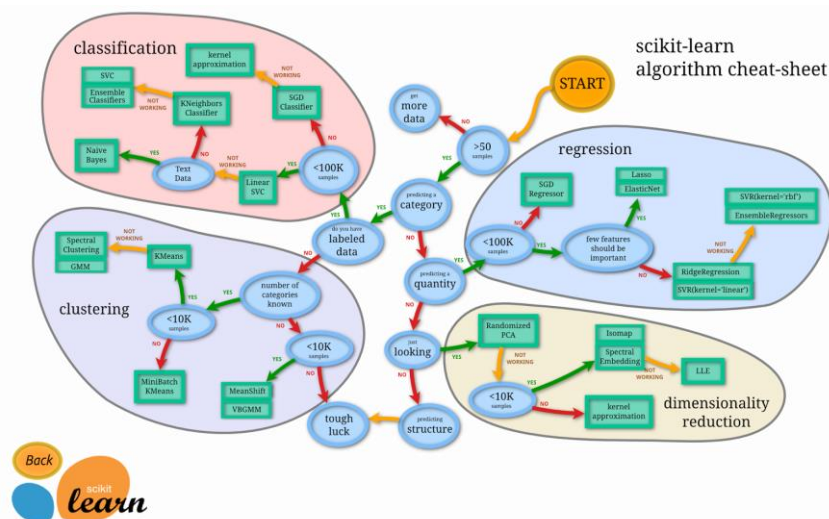


Figura 2: “Chuleta de selección de algoritmo” [5]

Los algoritmos de clasificación [6] son usados cuando el grupo de datos que manejamos es de tipo discreto, es decir, cuando los datos de salida (las respuestas) se comprenden entre un número finito de posibilidades, a diferencia de los algoritmos de regresión, que son útiles para predecir resultados de tipo continuo.

Si las posibles respuestas del algoritmo de clasificación son dos clases, por ejemplo “sí” y “no”, diremos que la clasificación es binaria. Un ejemplo de clasificación binaria [7] podría ser la detección de spam en un sistema de correo electrónico, donde la clasificación puede ser “spam” o “no spam”.

Sin embargo, si tenemos que predecir más de dos clases diremos que la clasificación es multicategoría. Un ejemplo de clasificación multicategoría podría ser el reconocimiento de letras en un escrito, en cuyo caso la clasificación tendrá tantas posibles salidas como letras tiene el abecedario. Diferenciar entre estos dos tipos de algoritmos de clasificación resulta útil para este TFG porque utilizaremos distintos tipos de clasificaciones según la fase del trabajo en la que se emplean cada uno de los algoritmos.

Los algoritmos de CL [8] son aquellos que tienen cuya finalidad realizar una agrupación de los datos de entrada en conjuntos según su similitud. Los datos que forman un conjunto se parecen más entre sí (en base a algún criterio determinado) que los datos de otras agrupaciones.

Un algoritmo de CL es el K-Medias (en inglés *K-Means*, KM) que consiste en agrupar los datos en un número de agrupaciones K, según su distancia a los centroides de cada conjunto de datos. Se parte de unos centroides generados aleatoriamente y en cada iteración el dato correspondiente será asignado a la agrupación cuyo centroide se encuentre más cerca, y posteriormente el centroide será recalculado según las distancias a cada muestra dentro del conjunto. El valor de K que funciona mejor en KM depende de las muestras que se manejan por lo cual no se conoce en un principio.

2.5 Coeficiente de silueta

El coeficiente de silueta [9] (en inglés *Silhouette*) es una herramienta usada para determinar el número de agrupaciones óptimas en un algoritmo de CL. Mide la calidad de los agrupamientos y resulta útil en algoritmos de aprendizaje no supervisado como KM, ya que este algoritmo no determina el número óptimo de agrupamientos.

El coeficiente de silueta se calcula de la siguiente manera:

$$s(i) = \frac{b-a}{\max(a,b)} \quad (5)$$

La variable ‘a’ determina el promedio de las distancias entre la muestra ‘i’ con las demás muestras del agrupamiento al que está asignado. La variable ‘b’ representa la distancia mínima que hay desde la muestra ‘i’ hasta el agrupamiento más cercano distinto del que ya está asignado.

Los posibles resultados de la fórmula anterior son los siguientes:

$$s(i) = \begin{cases} 1 - \frac{a}{b}, & \text{si } a < b \\ 0, & \text{si } a = b \\ \frac{b}{a} - 1, & \text{si } a > b \end{cases} \quad (6)$$

Sabiendo que el valor del coeficiente de silueta varía entre -1 y 1 ($-1 \leq s(i) \leq 1$), analicemos los posibles resultados:

- Para un valor de silueta cercano a 1, 'a / b' debe ser menor que 1 lo que significa que 'a' debe ser menor que 'b' ($a \leq b$). Esta desigualdad representa que la distancia desde la muestra 'i' a los demás agrupamientos es suficientemente grande como para afirmar que está bien asignada a su agrupamiento. Cuanto mayor sea el valor del coeficiente de silueta, mejor estará adaptada la muestra al agrupamiento que se ha asignado.
- Un valor de 0 implica que 'a = b', lo cual se traduce en que la muestra 'i' está exactamente entre dos agrupamientos.
- Si el coeficiente de silueta resulta en un valor cercano a -1, el valor de 'b' será mayor que el valor de 'a' y representa que la distancia desde la muestra 'i' a otros agrupamientos es suficientemente pequeña como para afirmar que está asignada a un agrupamiento que no es el correcto. Cuanto menor sea el valor del coeficiente, es decir, cuanto más cercano a -1 se encuentre peor estará la muestra 'i' asignada a su agrupamiento, por lo tanto, su pertenencia a otro agrupamiento será más evidente.

El gráfico de silueta reúne los coeficientes de silueta de todas las muestras analizadas y se puede interpretar como un gráfico de barras horizontales. En la siguiente figura expondremos un ejemplo del gráfico de silueta:

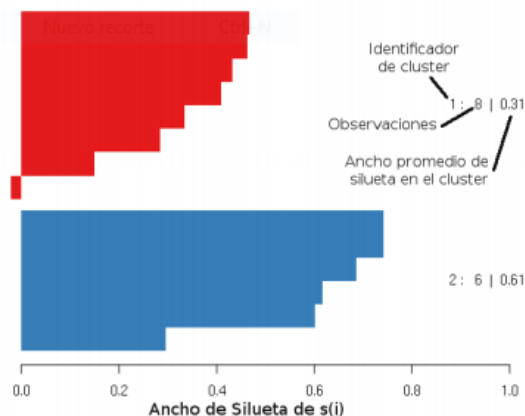


Figura 3: Gráfico de silueta. [9]

Cada barra del eje de ordenadas del gráfico representa una muestra del conjunto de datos analizados, y su correspondiente valor del coeficiente de silueta viene mostrado en el eje de abscisas.

En este ejemplo se han agrupado las muestras en dos conjuntos representados en el gráfico mediante el color rojo y el color azul. En la agrupación de color rojo se observa que hay una muestra con coeficiente de silueta negativo, lo cual indica que no está bien asignada a su conjunto. Por lo tanto, tendremos que hacer más pruebas utilizando un número de agrupaciones distinto a 2 para encontrar el número de conjuntos ideales.

2.6 Conclusiones

Los ataques DoS presentan una gran amenaza para nuestros sistemas, por lo tanto, detectarlos correctamente es muy importante. Si hacemos uso de las características y de los parámetros de la distribución α -estable, es posible desarrollar un algoritmo que, configurado de manera óptima, pueda llegar predecir el ataque haciendo uso de la distribución α -estable

3 Diseño

3.1 Introducción

Este capítulo está dedicado a explicar las decisiones de diseño llevadas a cabo durante todo el proceso del trabajo, así como los motivos por los cuales han sido tomadas.

El diseño de este TFG se divide en los siguientes puntos:

- MATLAB
- Etiquetas de clasificación
- Algoritmo de CL
- Coeficiente de Silueta

3.2 MATLAB

Al comenzar a plantear al trabajo, hubo que tomar una decisión para determinar en qué software sería desarrollado el código. Hubo varios aspectos a considerar, entre los cuales estarían: los conocimientos previos del lenguaje de programación, el lenguaje en el que estaba desarrollado el código del que parte este TFG, los algoritmos a desarrollar, así como el prototipado de las funciones que deberíamos utilizar.

Los posibles softwares o lenguajes de programación candidatos para ser utilizados fueron MATLAB, C y Python. A continuación, se dispone de un esquema de tipo semáforo que resume los mejores y los peores aspectos considerados de cada uno:






















	 MATLAB	 C	 Python
Código de partida			
Conocimientos previos			
Implementación de algoritmos			
Repositorios de funciones			
Manejo de matrices de datos			
Representaciones gráficas			

Figura 4: Comparativa de lenguajes.

Es importante aclarar que en cada categoría se usa el código de colores forma relativa, esto significa que un color rojo no tiene por qué significar deficiencia, sino que está por debajo en comparación a los otros.

Como se puede ver las mejores opciones eran MATLAB o Python, por lo que C quedó descartado desde un principio. Un factor determinante para la elección fue que, tanto el código como los datos de partida obtenidos de los TFG anteriores estaban desarrollados en extensiones compatibles con MATLAB. Esto combinado con un conocimiento previo del lenguaje y con la utilidad que nos proporcionaba MATLAB para desarrollar el trabajo propuesto mediante su amplia librería de ML, fueron las razones por las que fue elegido.

3.3 Etiquetas de clasificación

El tráfico analizado fue proporcionado por el TFG de partida y se compone de dos series temporales: la serie de tráfico real, obtenida gracias a la monitorización del tráfico de red de la Universidad de Granada durante 7 días, y una serie de tráfico sintético que representaba un ataque DoS, también proporcionada por la Universidad de Granada. La serie temporal analizada es una mezcla de estas dos series temporales, tráfico real más ataque sintético.

Para extraer los parámetros α -estables de la serie temporal es necesario utilizar ventanas temporales [10]. Estas ventanas se deslizan segundo a segundo realizando el ajuste de la distribución alfaestable para los valores dentro de la ventana.

Los cálculos fueron realizados, en un principio, con una ventana de duración de 5 minutos ya que en el TFG anterior se llegó a la conclusión de que esta ventana conseguía extraer los parámetros α -estables de forma que se pudiese diferenciar las zonas de tráfico de ataque y tráfico normal. Posteriormente se varió la duración de la ventana y comprobamos que efectivamente, los mejores resultados se conseguían con la ventana de 5 minutos.

Para clasificar los tipos de ataque es necesario utilizar etiquetas. En el TFG de partida se realizaba una clasificación binaria del tráfico, distinguiendo entre dos tipos: tráfico normal, que se corresponde con el fragmento de la serie temporal que no ha sido alterado por el ataque; y tráfico de ataque, que se compone por la mezcla entre el ataque informático y el fragmento de tráfico normal donde actúa el ataque.

Sin embargo, en este trabajo se exploró la posibilidad de cambiar la clasificación a una multicategoría, añadiendo una etiqueta más que representa el tráfico mixto. Este tráfico mixto se corresponde con aquel fragmento de la serie temporal donde la ventana deslizante recoge datos simultáneamente tanto de tipo normal como de tipo ataque.

La siguiente figura muestra las tres etiquetas consideradas para la clasificación según la posición de la ventana temporal:

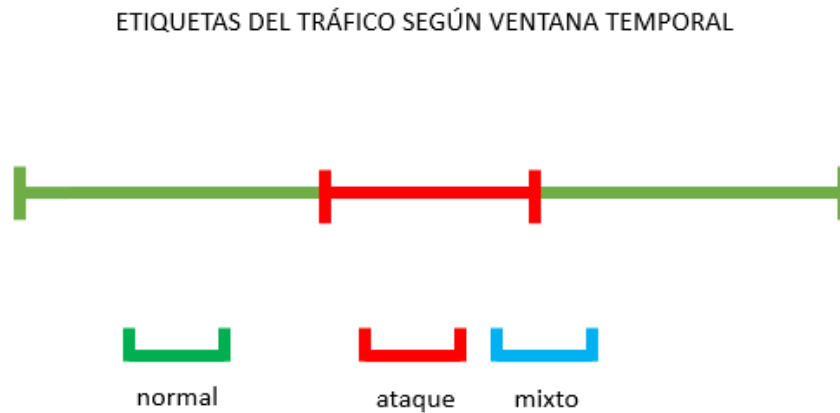


Figura 5: Tipos de tráfico.

El objetivo de incluir una etiqueta de tráfico mixto fue ver si aportaba algún tipo de información e investigar si el ataque también pudiese ser predecible antes de que la ventana temporal estuviese completamente dentro del fragmento de tráfico de ataque.

Viendo el esquema anterior vemos como el tráfico mixto se origina en dos puntos:

- Cuando la ventana pasa de tráfico normal a tráfico de ataque
- Cuando la ventana pasa de tráfico de ataque a tráfico normal.

Por lo tanto, dentro del tráfico mixto consideraremos dos tipos de tráfico: tráfico mixto antes del ataque al que nos referiremos en las gráficas como 'mixtopre' (mixto pre-ataque) y tráfico mixto después del ataque, al cual nos referiremos como 'mixtopost' (mixto post-ataque).

Esta distinción fue realizada no sólo para ver si las zonas en las que caían sus parámetros alfaestables fuesen claramente diferenciables, sino porque predecir el tráfico mixto pre-ataque aportaba una utilidad mayor que predecir el tráfico mixto post-ataque, dado que este se generaba una vez ya que el ataque había sido efectuado.

3.4 Algoritmo de CL

Para implementar el algoritmo de agrupación sobre los parámetros α -estables fue necesario buscar cuál de ellos se adaptaba mejor a nuestras condiciones. Primeramente se probó con la función `clusterdata()` [11] incorporada en la librería de ML de MATLAB. Esta función construye agrupaciones aglomerativas a partir de los datos de entrada.

Su algoritmo funciona de la siguiente manera:

- 1 – Calcula la distancia entre pares de muestras
- 2 – Crea un árbol jerárquico aglomerativo computando la distancia mínima entre agrupaciones
- 3 – Crea las agrupaciones a partir de los datos del árbol jerárquico mediante la función `cluster()`.

Esta función tuvo que ser descartada dado que devuelve los índices de las agrupaciones y no devuelve las coordenadas de los centroides, que son los puntos que necesitamos para construir la clasificación.

La función elegida para la agrupación fue `kmeans()` [12] ya que esta sí permite conocer la información de los centroides que genera. Además, este algoritmo nos permite jugar con la distancia definida en el espacio, para ver con cual de ellas obteníamos mejores resultados, y nos da la opción a elegir el número de centroides.

Las posibles distancias que admite la función de KM de MATLAB son las siguientes: distancia euclídea cuadrada, distancia de Hamming, distancia ‘cityblock’, distancia del coseno y distancia de correlación. La distancia de Hamming fue descartada ya que sólo funciona para muestras binarias. Tras realizar diferentes pruebas con cada una de ellas, la que mejor conseguía calcular los centroides para la clasificación posterior fue la distancia euclídea cuadrada, ya que los centroides que calcula en cada agrupación son la media de los puntos en dicha agrupación, lo cual resultaba eficaz a la hora de clasificar los datos.

3.5 Coeficiente de silueta

El coeficiente de silueta fue una herramienta de diseño utilizada para configurar el algoritmo de CL, dado que KM no nos calcula el número de agrupaciones correcto para nuestros datos. El objetivo de su uso fue determinar el número de centroides óptimo para cada tipo de tráfico de tal forma que quedasen correctamente situados, para posteriormente utilizarlos en la clasificación.

El proceso para trabajar con los coeficientes de silueta fue el siguiente:

- Se escogió el mínimo número de agrupaciones para cada tipo de tráfico, es decir, 2, para calcular el coeficiente de silueta.
- Se ejecutó el algoritmo de CL y se obtuvo el gráfico de silueta.
- Si los coeficientes de silueta no tienen valores negativos y son suficientemente grandes, se adopta dicho número de agrupaciones como el número óptimo.

- Se vuelve a realizar el proceso esta vez con incrementando en 1 el número de centroides. El proceso se itera hasta que se hayan encontrado los números óptimos de agrupaciones para cada tipo de tráfico o en su defecto, hasta que los gráficos de silueta comiencen a representar coeficientes cada vez menores, en cuyo caso nos habremos pasado del número óptimo de agrupaciones.
- Si no se ha encontrado el número óptimo de agrupaciones de algún tipo de tráfico, analizamos los resultados de todos los gráficos y elegimos el que contenga menor número de coeficientes de silueta negativos.

3.6 Conclusiones

En este capítulo se han revisado todas las decisiones de diseño tomadas antes y durante el desarrollo del trabajo:

- Se optó por utilizar el entorno de MATLAB para desarrollar todo el código gracias a la utilidad que aportaba frente a otros lenguajes de programación.
- Se definieron las etiquetas de los distintos tipos de tráfico que se querían clasificar, siendo estos cuatro: tráfico normal, tráfico de ataque, tráfico mixto pre-ataque y tráfico mixto post-ataque.
- Se decidió utilizar KM con una distancia euclídea cuadrada como el algoritmo de CL.
- Y finalmente, se utilizó el coeficiente de silueta para definir el número de agrupaciones por cada tipo de tráfico en el algoritmo de CL.

4 Desarrollo

4.1 Introducción

En este capítulo se expone el proceso que se ha llevado a cabo para desarrollar el trabajo, explicando detalladamente cada uno de los pasos seguidos al igual que las funciones utilizadas, los ajustes realizados y algunos de los resultados obtenidos.

En el siguiente diagrama de bloques se detalla el transcurso del trabajo, identificando las fases por las que se ha pasado. Los bloques en color verde indican los componentes desarrollados en el presente TFG:

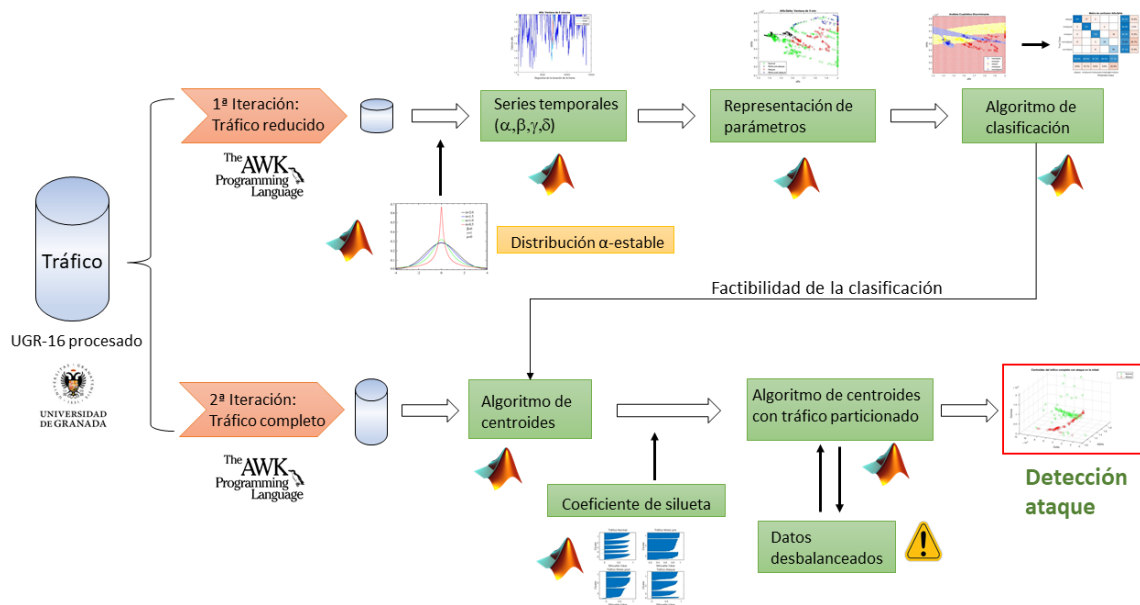


Figura 6: Diagrama de bloques del desarrollo.

4.2 Series temporales

En esta primera iteración del trabajo se cogieron las dos series temporales de tráfico, la de tráfico normal y la de ataque, fueron introducidas en MATLAB mediante un fichero '.mat' que contenía dos columnas, una dedicada al tiempo UNIX en la que se había desarrollado la monitorización del tráfico, y otra que contenía los datos del flujo de red medidos en número de paquetes. Se escogió la cantidad medida en paquetes en vez de en bits ya que en el TFG de partida se había observado como los parámetros α -estables obtenidos del tráfico en bits eran similares para las zonas de tráfico de ataque y las zonas de tráfico normal.

El primer paso que se dio en el trabajo fue representar los parámetros α -estables que se habían obtenido del TFG anterior, obtenidos de segmento del tráfico total mediante una ventana de 5 minutos. El tráfico reducido que se había analizado suponía una fracción de 1/42 sobre el tráfico completo.

El objetivo de esta primera tarea era analizar cada uno de los parámetros α -estables en una representación temporal para ver qué parámetros podrían aportar más información.

Se realizó una gráfica por cada parámetro alfaestable y en cada una de ellas se representaron dos series temporales: el tráfico real sin ataque y el tráfico real mezclado con el ataque. En la serie temporal que contiene el ataque se distinguen las zonas de tráfico de ataque y tráfico mixto. Las series temporales duraban 14101 segundos, y el ataque estaba situado aproximadamente en el centro.

A continuación, se muestran las series temporales de los parámetros obtenidas mediante la representación en MATLAB:

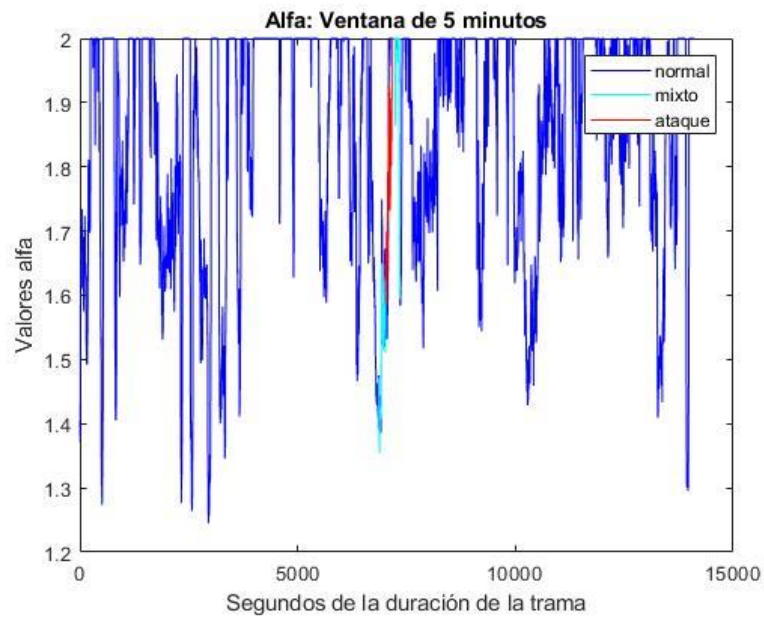


Figura 7: Serie temporal de Alfa.

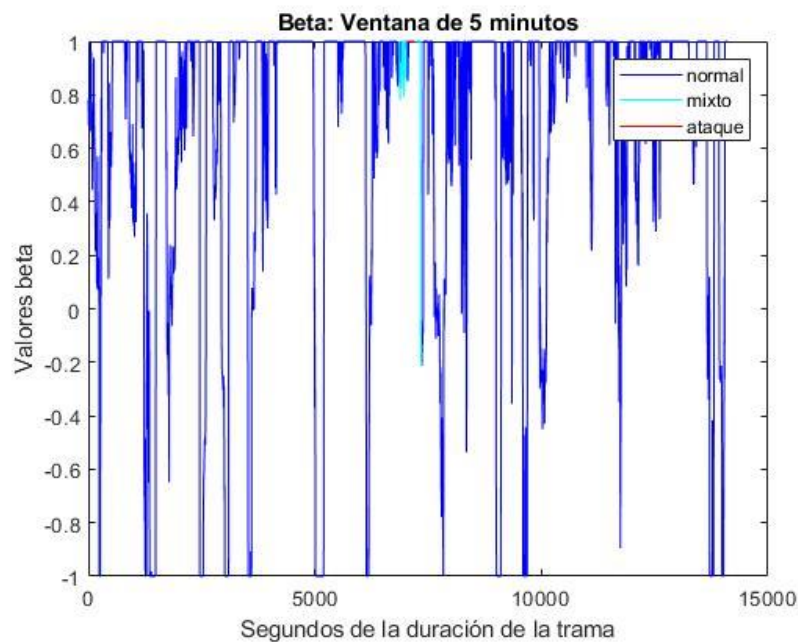


Figura 8: Serie temporal de Beta.

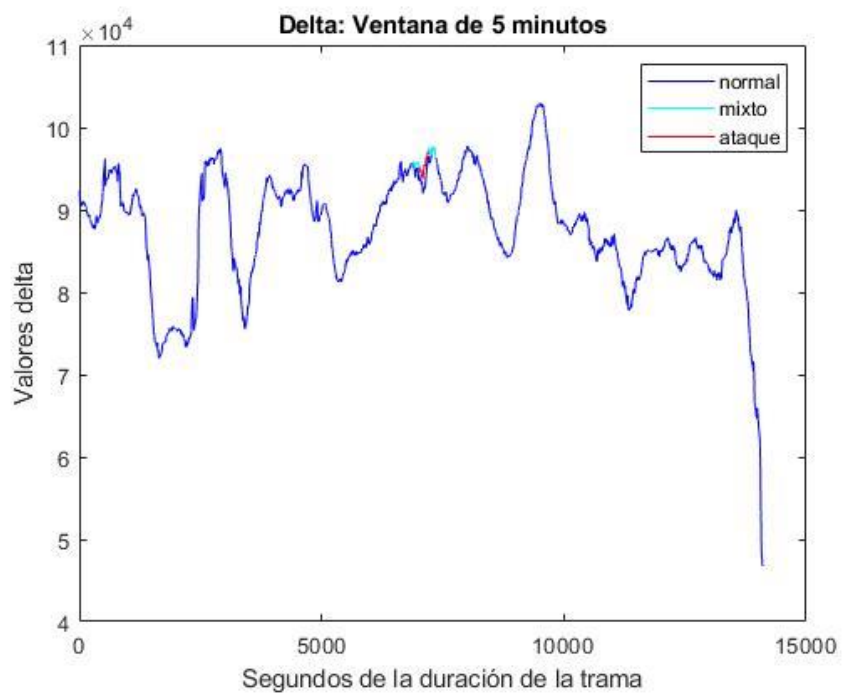


Figura 9: Serie temporal de Delta.

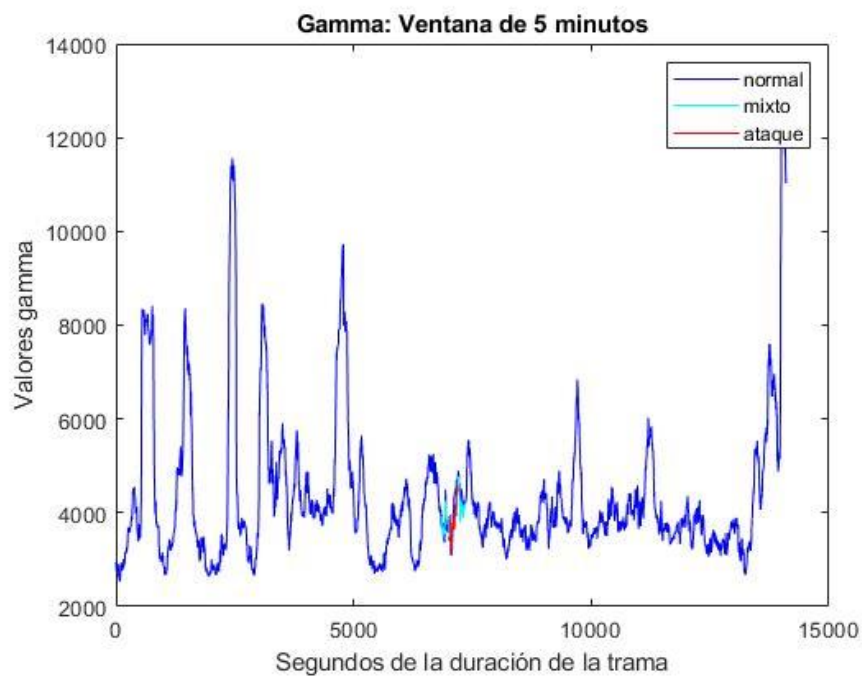


Figura 10: Serie temporal de Gamma.

Se utilizan tres colores para distinguir los tres tipos de tráfico: azul oscuro para el tráfico normal, rojo para el tráfico de ataque y cian para el tráfico mixto (el mismo color para el tráfico mixto pre-ataque y tráfico mixto post-ataque).

En las representaciones de alfa y beta la zona de ataque se diferencia muy poco respecto a la zona de tráfico normal. En la representación de gamma se puede distinguir un poco mejor la zona de ataque, pero era en la representación de delta en la que se apreciaba claramente un orden y una separación entre todos tipos de tráfico.

En la siguiente gráfica se observa la representación de delta, pero esta vez, enfocando la zona de ataque para apreciar mejor la separación entre las zonas de los distintos tipos de tráfico:

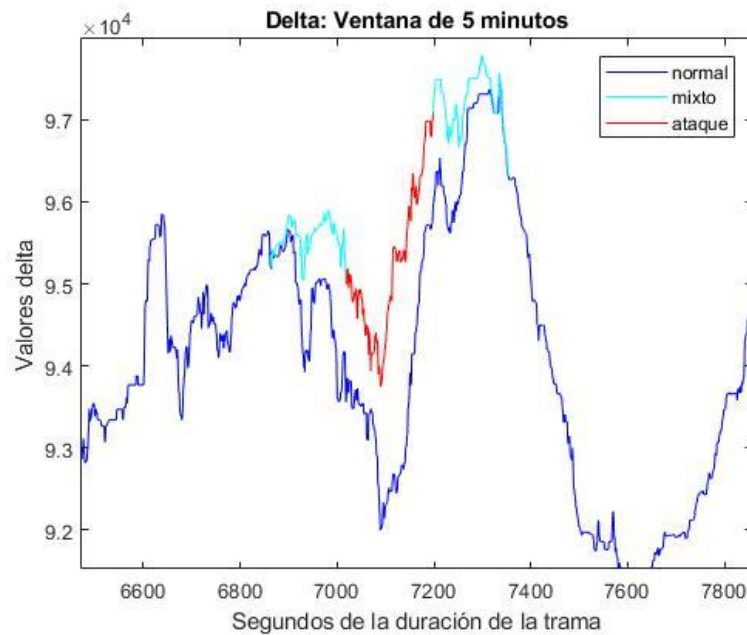


Figura 11: Serie temporal de Delta en la zona del ataque.

En el gráfico anterior se distinguen claramente los tres tipos de tráfico y sus transiciones, ilustrando el funcionamiento de las ventanas temporales. La trama comienza con el tráfico normal al que le sigue el tráfico mixto pre-ataque en el segundo 6862. En ese momento observamos como el tráfico normal continua por debajo de la zona de ataque, que es precisamente el recorrido que hubiese adoptado alfa si la red no se hubiese visto atacada. En el segundo 7018 el tráfico mixto pasa a tráfico de ataque el cual perdura hasta el segundo 7198, produciéndose la transición a tráfico mixto post-ataque. Finalmente, el tráfico mixto se reúne con el tráfico normal concluyendo así el ataque.

Tras evaluar las series temporales se concluye que delta podría ser una variable clave para construir la clasificación, pero para corroborarlo se analizarán otros aspectos aparte del temporal.

4.3 Representaciones de los parámetros

Tras analizar el aspecto temporal de cada parámetro α -estable, el siguiente paso que se dio para obtener más información acerca de ellos fue representarlos uno frente a otro en un espacio bidimensional, situando un parámetro en el eje de las abscisas y otro en el eje de las ordenadas. Por tanto, cada punto del espacio es representado mediante los dos parámetros en cuestión.

De esta forma, se obtuvo un total de 6 representaciones dentro de las cuales se distinguieron los tres tipos de tráfico. El objetivo de las representaciones era observar que valores tomaban los parámetros en cada tipo de tráfico, de tal forma que pudiésemos sacar algún tipo conclusión acerca de qué parámetro nos aportaría más información.

A continuación, se exponen las 4 de las 6 representaciones, omitiendo dos de las representaciones de beta (alfa-beta y beta-gamma) dado que no proporcionan información adicional.

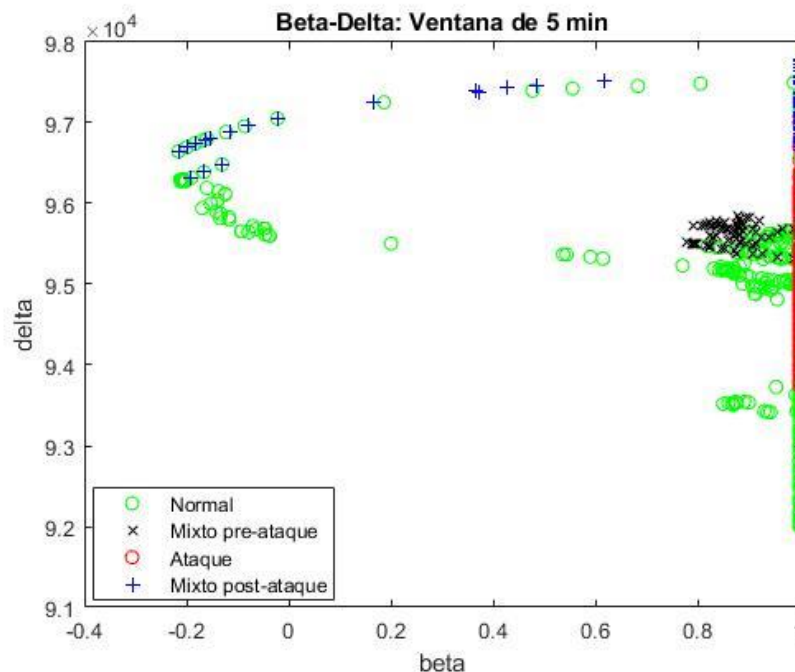


Figura 12: Representación de Beta frente a Delta.

En esta gráfica de beta frente delta se observa que los valores beta tienden a uno, situándose la mayoría de las muestras en este valor o cerca de él. Los valores de beta del tráfico de ataque se sitúan todos en 1, lo cual implica que estos valores de ataque no se diferencien de otros valores de beta de distinto tipo de tráfico y por tanto beta no sirve para el algoritmo de CL. Por tanto, beta quedó descartada ya que no aportaba mucha información.

Sin embargo, beta sí podría ser de utilidad para clasificar el tráfico de ataque introduciendo una condición en el algoritmo de clasificación que comprobase si el valor beta del punto clasificado como ataque es igual a 1.

Ahora se muestran las gráficas que contienen los otros parámetros:

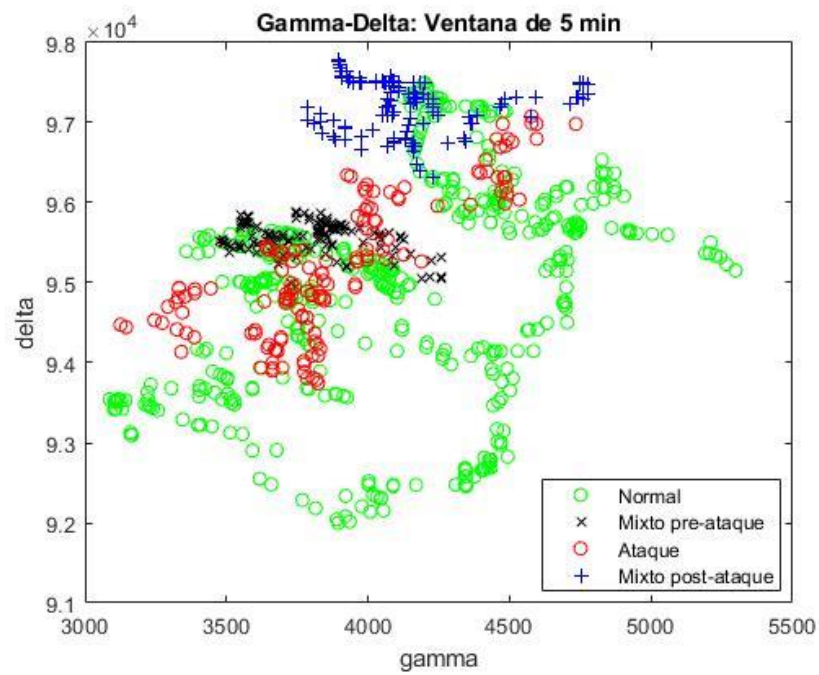


Figura 13: Representación de Gamma frente a Delta.

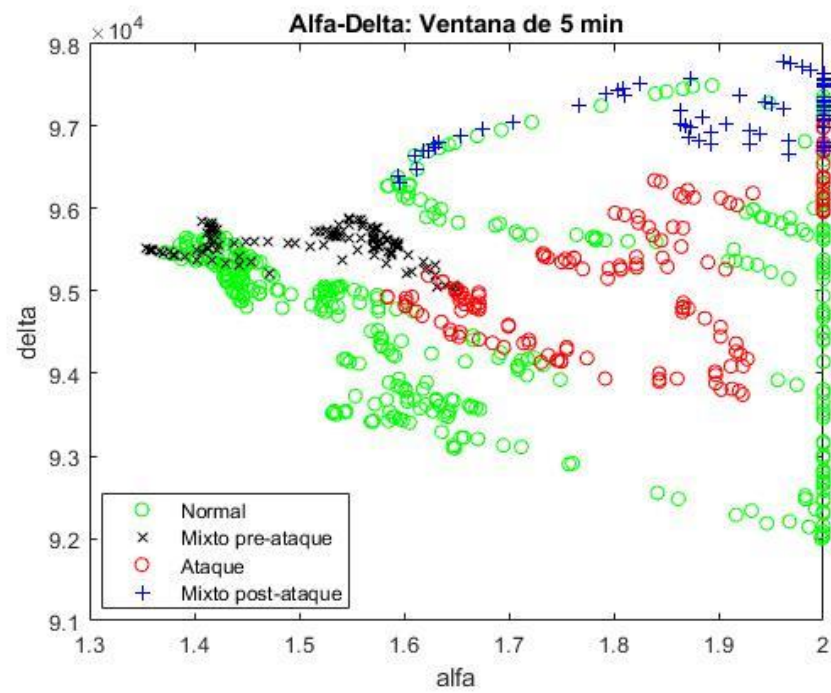


Figura 14: Representación de Alfa frente a Delta.

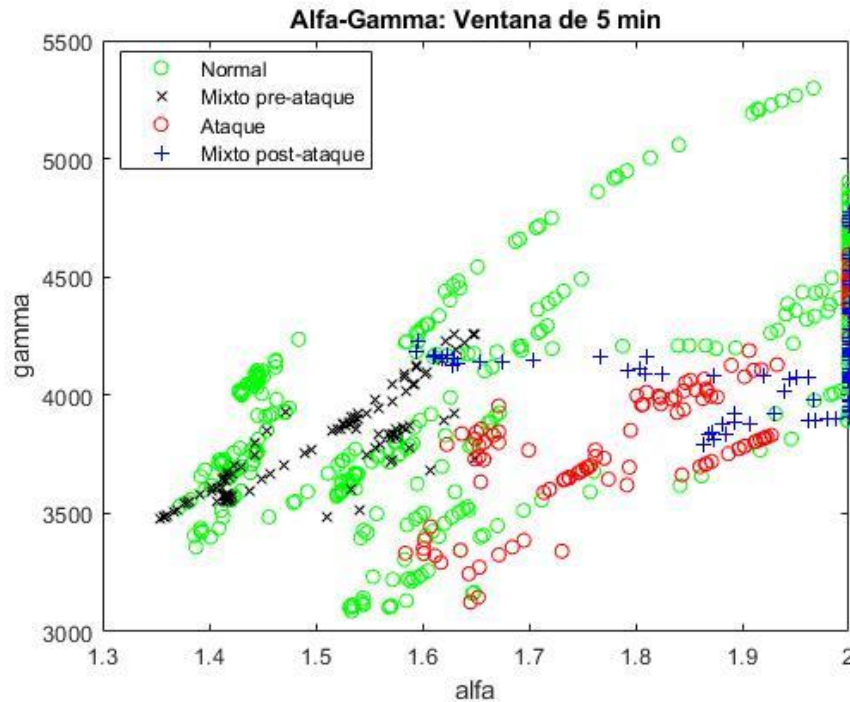


Figura 15: Representación de Alfa frente a Gamma.

En estas representaciones, por el contrario, sí se pueden distinguir mejor las zonas de los diferentes tipos de tráfico. En aquellas que contienen el parámetro alfa podemos observar una concentración de las muestras entorno al valor de alfa igual a 2. Pero, al contrario que lo que ocurría con la gráfica de beta, se tienen más muestras de todos los tipos de tráfico a lo largo de los distintos valores de alfa, lo cual permite diferenciar mejor cada zona de tráfico.

En la representación de gamma frente a delta se observa una distribución de las muestras parecida a la vista anteriormente, sin embargo, no existe un valor concreto de uno de los dos parámetros en el que se aglomeren los datos.

Observando las tres gráficas, la que mejor separa las distintas zonas de tráfico es la representación de alfa frente a delta, ya que sus muestras de distintas clases de tráfico se mezclan menos entre sí que en las otras dos gráficas, por lo tanto, es posible que sea más útil a la hora de implementar nuestro algoritmo de clasificación.

4.4 Algoritmo de clasificación

Una vez se han observado las representaciones de los parámetros y se han analizado cuales de ellas podrían aportar más información, se construye el algoritmo de clasificación y se prueba su efectividad. El algoritmo fue programado en un script de MATLAB en el cual se implementó un análisis cuadrático discriminante mediante la función *classify()* para cada representación obtenida en el apartado anterior.

Lo primero fue construir el clasificador que consiste en un array de la misma longitud que la trama de datos. Dicho array contiene números que representan las distintas etiquetas de tráfico, de tal forma que el primer tipo de tráfico (tráfico normal) esta asociado con el número 1 y se repetiría tantas veces como observaciones de ese tipo de tráfico haya hasta el momento que se produzca la transición a la siguiente clase de tráfico (mixto pre-ataque), el cual pasaría a representarse con un 2. Esto se repetiría sucesivamente hasta tener un array de unos, doses, treses, cuatros y cincos, con cada cifra asociada a una clase de tráfico distinta.

El siguiente paso fue hacer uso de la función *meshgrid()* para formar una cuadrícula 2-D que se adaptase a cada uno de los parámetros representados. Los valores de alfa estaban comprendidos entre 1.3 y 2, delta entre 9×10^4 y 10×10^4 y gamma entre 3×10^3 y 6×10^3 . Para compensar la diferencia de magnitud en la cuadrícula se estableció un paso de 0.01 para la variable alfa y un paso de 100 para gamma y delta, de esta forma se tiene un número similar de puntos en cada eje del espacio.

Seguidamente, se llamaría a la función *classify()* a la cual se le pasarían los siguientes argumentos:

- La matriz que describe el espacio formado mediante la cuadrícula (la salida de la función *meshgrid()*).
- La matriz de los datos de entrenamiento, cuyas tres columnas representarían los tres parámetros α -estables analizados (alfa, delta y gamma).
- El clasificador construido mediante el array de etiquetas numéricas.
- Un parámetro que especificaría el tipo de análisis a implementar. Las posibles opciones son: 'linear', 'quadratic', 'diaglinear', 'diagquadratic', y 'mahalanobis'. Se escoge 'quadratic' para realizar un análisis discriminante cuadrático ya que vemos en las representaciones extraídas del apartado 4.3 que un análisis lineal no sería efectivo y tras realizar pruebas con el cuadrático se observa que es el que mejor se adapta a las observaciones.

De esta forma se obtuvieron tres representaciones similares a las anteriores pero cuyo espacio esta dividido en regiones de distintos tipos de tráfico. A continuación, se muestran dichas gráficas:

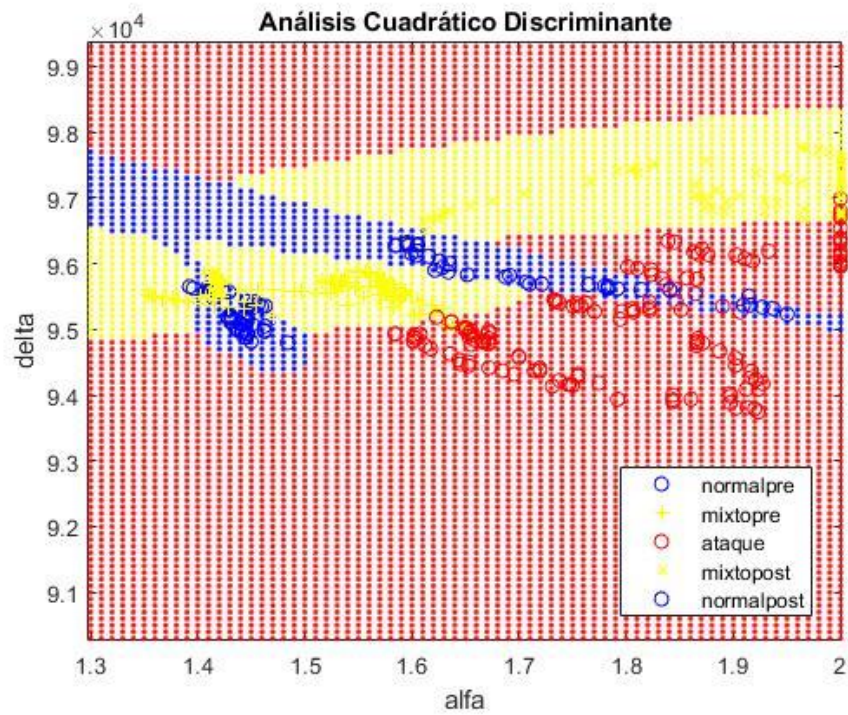


Figura 16: Clasificación Alfa frente a Delta

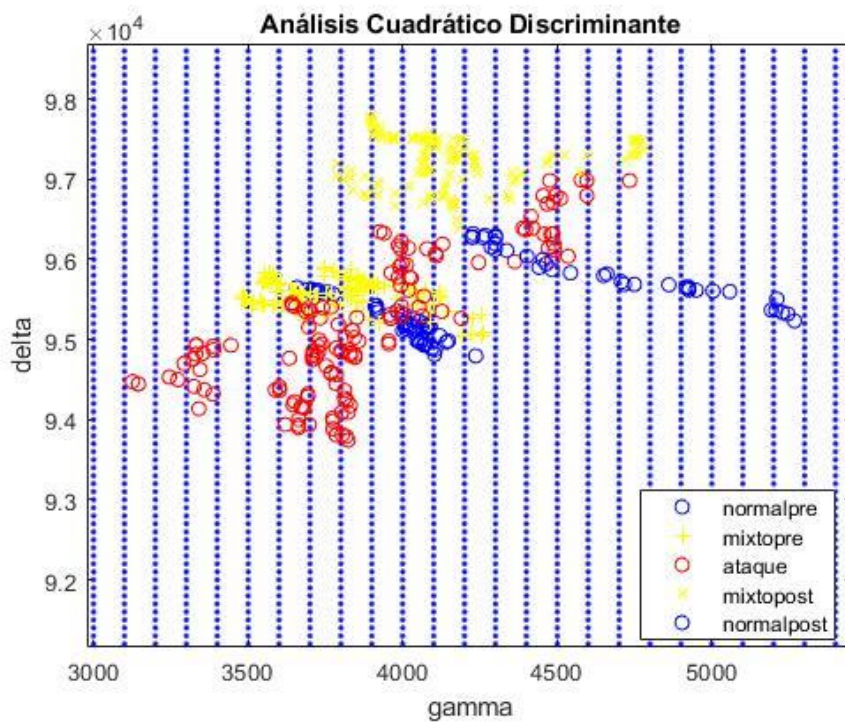


Figura 17: Clasificación de Gamma frente a Delta.

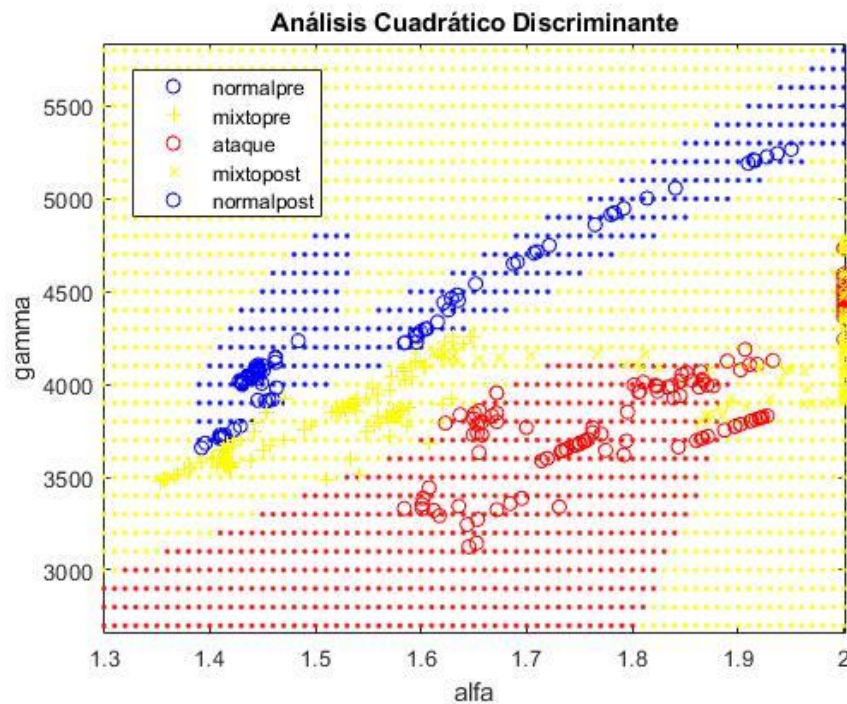


Figura 18: Clasificación de Alfa frente a Gamma.

Como antes se había predicho, la gráfica en la que mejor sale la clasificación del espacio es en alfa frente a delta y en alfa frente a gamma. Por el contrario, la función *classify()* no consigue hacer una clasificación correcta en la gráfica de delta frente gamma.

Adicionalmente, se propuso realizar una clasificación de tres dimensiones con los tres parámetros alfa estable la cual resultó de gran utilidad para visualizar gráficamente las observaciones de los tres parámetros conjuntamente, pero no se pudo realizar la clasificación del espacio tridimensional debido a las limitaciones de las funciones representación de MATLAB como *scatter3()* y *plot3()*.

Para finalizar esta sección se observarán las matrices de confusión obtenidas a partir de la clasificación realizada con las funciones *fitcdiscr()*, que realiza un modelo clasificador del análisis discriminante y *resubPredict()*, que genera una predicción respecto a dicho modelo de clasificación. El proceso para implementarlas es el siguiente:

- Primero se llama a la función *fitcdiscr()*, a la cual se le pasan los datos a clasificar, un array similar al clasificador explicado anteriormente pero, en vez de utilizar números para etiquetar el tráfico, con las etiquetas en forma de caracteres.
- Por último, se le pasa como parámetro '*quadratic*' para que realice el ajuste de forma cuadrática ya que hemos visto que es la más apropiada.
- La salida de la función anterior es introducida después como entrada a la función *resubPredict()* con la cual obtenemos los datos clasificados según la predicción.

Con las matrices de confusión se comparan las etiquetas de los datos obtenidas con la predicción y las etiquetas reales, estableciendo porcentajes de acierto y porcentajes de fallo en cada clase de tráfico para ver la efectividad de la predicción. Se realizarán en total 4 matrices de confusión: 3 para las observaciones de los pares de parámetros vistos en las gráficas anteriores (Alfa-Delta, Alfa-Gamma, Delta-Gamma) y otra para las observaciones de los tres parámetros conjuntamente (Alfa-Delta-Gamma). La que mejor resultados dio de las 3 primeras fue Alfa-Delta, que se muestra a continuación:

		Matriz de confusion Alfa-Delta						
True Class	ataque	161	17	2			89.4%	10.6%
	mixtopost	3	151		2		96.8%	3.2%
	mixtopre	5		135		16	86.5%	13.5%
	normalpost	9		3	34		73.9%	26.1%
	normalpre			8		54	87.1%	12.9%
		90.4%	89.9%	91.2%	94.4%	77.1%		
		9.6%	10.1%	8.8%	5.6%	22.9%		
		ataque	mixtopost	mixtopre	normalpost	normalpre		
		Predicted Class						

Figura 19: Matriz de confusión Alfa-Delta.

Esta clasificación obtiene un acierto en las muestras de ataque reales de casi el 90%, con unos falsos positivos de 9.6% en comparación al 70.6% de acierto de Delta-Gama y el 65% de Alfa-Gamma. Pero la que mejor resultados ofreció fue la matriz de confusión Alfa-Delta-Gamma, lo cual es razonable ya que utiliza las observaciones de los tres parámetros, por lo tanto, la clasificación resultante es más precisa. A continuación, se muestra dicha matriz de confusión:

		Matriz de confusion Alfa-Gamma-Delta						
True Class	ataque	168	12				93.3%	6.7%
	mixtopost		153		3		98.1%	1.9%
	mixtopre	3		153			98.1%	1.9%
	normalpost				46		100.0%	
	normalpre					62	100.0%	
		98.2%	92.7%	100.0%	93.9%	100.0%		
		1.8%	7.3%		6.1%			
		ataque	mixtopost	mixtopre	normalpost	normalpre		
		Predicted Class						

Figura 20: Matriz de confusión Alfa-Gamma-Delta.

Esta matriz de confusión es notablemente mejor que las anteriores obteniendo un 93.3% de porcentaje de acierto en la clasificación del tráfico de ataque real con solamente un 1.8% de falsos positivos, y consiguiendo clasificar perfectamente el tráfico normal. Esta clasificación demuestra que, haciendo uso de los parámetros α -estables alfa, gamma y delta, es posible clasificar de forma efectiva el tipo de tráfico de una trama reducida. Con estos resultados se termina de analizar el tráfico reducido y pasaríamos a trabajar con la trama completa de tráfico.

4.5 Algoritmo de clasificación mediante centroides

Tras realizar las matrices de confusión se comienza a trabajar con la trama de tráfico completa generando un algoritmo que extrajese los parámetros α -estables de toda la trama y los clasificase según su proximidad a los centroides de cada tipo de tráfico. El funcionamiento del algoritmo de MATLAB que se generó fue el siguiente:

- Primero, se genera un script que permita mezclar el tráfico normal y el de ataque de forma que se pueda posicionar el ataque en la posición que se desee.
- Seguidamente, tras obtener el tráfico mezclado (tráfico normal más tráfico de ataque), se le pasa la ventana de 5 minutos extrayendo punto a punto los parámetros α -estables y separándolos en distintas matrices de tres columnas según el tipo de tráfico al que pertenecen, tal y como se explico en el apartado 3.3. Cada una de las tres columnas de dichas matrices es asignada a un parámetro alfaestable, por lo tanto, cada fila dentro de la matriz podrá ser interpretada como un punto en el espacio tridimensional alfa-gamma-delta.
- Se utiliza KM con la distancia euclídea cuadrada (como explicamos en el apartado 3.4) en cada matriz de datos para extraer los centroides de cada clase de tráfico, que también constarán de tres componentes (alfa, delta y gamma)
- Se calculan las distancias de todas las muestras a cada uno de los centroides mediante la función de MATLAB *pdist2()* utilizando una distancia euclídea cuadrada, y se realiza una clasificación cada muestra según el centroide más cercano: la etiqueta asignada a la observación estará determinada por el tipo de tráfico al que pertenezca el centroide más cercano.
- Se representan las gráficas del coeficiente de silueta para evaluar como de óptimas han sido las agrupaciones realizadas por KM para cada tipo de tráfico, mediante el proceso descrito en el apartado de diseño 3.5.
- Se representan los centroides y los datos obtenidos en una gráfica de tres dimensiones cuyos ejes son alfa, delta y gamma y se analizan los resultados de la clasificación mediante matrices de confusión similares a las vistas anteriormente.

Para conseguir un mejor funcionamiento del algoritmo hubo que realizar pruebas para configurar algunos de sus parámetros:

- Se probaron algunas de las distintas distancias que ofrecía la función *pdist2()* que son las siguientes: 'euclidean', 'squaredeuclidean', 'mahalanobis', 'hamming' y 'cityblock'. Pese a que los resultados eran muy similares entre el uso de la distancia euclídea cuadrada y la distancia de mahalanobis, la primera obtenía ligeramente mejores resultados por lo que fue escogida.

4.6 Coeficiente de silueta

Una vez escogida la distancia óptima, se ejecutó el algoritmo reiteradamente para obtener los coeficientes de silueta y determinar cuál era el número de centroides óptimo para cada clase de tráfico. Tras probar con más de 15 centroides por cada tipo de tráfico, se analizaron todas las gráficas del coeficiente de silueta y elegimos la cantidad de centroides que mejor se adaptan a nuestras muestras. La gráfica con los mejores coeficientes de silueta es la siguiente:

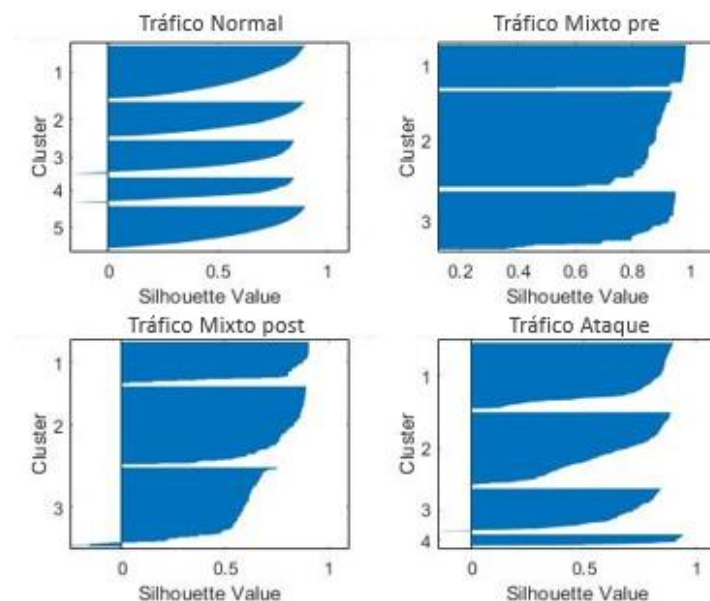


Figura 21: Gráficos de silueta con número óptimo de centroides.

El número de centroides óptimo que se observan en las gráficas son los siguientes:

- Tráfico normal = 5 centroides
- Tráfico mixto pre-ataque = 3 centroides
- Tráfico mixto post-ataque = 3 centroides
- Tráfico de ataque = 4 centroides

Como se observa en las gráficas de silueta anteriores, KM solo genera centroides que se adaptan perfectamente a las muestras de su agrupación para el tráfico mixto pre-ataque. Para los demás tipos de tráfico obtenemos algunos coeficientes de silueta negativos que implican que hay muestras que no están siendo agrupadas correctamente. Pero el número de estas muestras mal asignadas es el más pequeño que se podía conseguir, y en las muestras que están correctamente asignadas obtenemos valores del coeficiente de silueta altos, por eso se seleccionó estos números de centroides.

En el siguiente apartado analizaremos la matriz de confusión con los resultados de la clasificación del tráfico completo mediante este algoritmo.

4.7 Datos desbalanceados

El objetivo del algoritmo de clasificación es obtener una clasificación efectiva del tráfico total. El tráfico total tiene una duración de 605181 segundos mientras que el ataque duraba solamente 2169 segundos. Esto hacía que hubiese una proporción de 279 veces más de tráfico normal que de tráfico de ataque, lo cual hacía que los datos de la clasificación estuviesen muy desbalanceados. Este desbalanceo provocaba que muchos de los datos de ataque fuesen clasificados como datos de ataque debido a la inmensidad de puntos de tráfico normal.

La siguiente figura muestra la matriz de confusión de la clasificación realiza por el algoritmo en la que se puede ver este efecto:

Matriz de confusion Alfa-Gamma-Delta quadratic							
True Class	ataque	1140	43	507	181	60.9%	39.1%
	mixtopost	9	289			97.0%	3.0%
	mixtopre	16		274	8	91.9%	8.1%
	normal	47298	9705	84404	461007	76.5%	23.5%
		2.4%	2.9%	0.3%	100.0%		
		97.6%	97.1%	99.7%	0.0%		
	ataque	mixtopost	mixtopre	normal	Predicted Class		

Figura 22: Matriz de confusión con datos desbalanceados.

A la derecha de la figura se encuentra una tabla de dos columnas con las clases verdaderas. Se puede ver como el tráfico normal se ha predicho correctamente un 76.5 % de las veces. Sin embargo, el 23.5% restante se ha confundido con otro tipo de tráfico y supone que 47298 puntos de tráfico normal se hayan etiquetado como tráfico de ataque de forma errónea.

En la parte inferior de la figura se encuentra la tabla que nos muestra las clases predichas. Como se puede observar, los 47298 puntos de tráfico normal confundidos como tráfico de ataque suponen aproximadamente el 97.6% de los datos etiquetados como ataque, haciendo que solamente el 2.4% restante sean los datos de ataque predichos correctamente. Este es el problema de los datos desbalanceados: debido a la diferencia entre el número de datos de tráfico normal y los demás tipos de tráfico, se tiene un algoritmo que hace imposible detectar un ataque de forma efectiva ya que obtendrá una cantidad muy elevada de falsos positivos.

Lo mismo ocurre para el tráfico mixto pre-ataque y para el tráfico mixto post-ataque, siendo en este caso aún más acentuado el efecto ya que hay menos número de datos de tráfico mixto.

Para solucionar este problema se aplicaron dos mecanismos de mejora:

- Se cambió la estrategia de clasificación del algoritmo de centroides
- Se decidió particionar el tráfico normal en 10 segmentos, y estudiar el comportamiento del ataque cuando lo mezclábamos con cada uno de los segmentos del tráfico para así disminuir la proporción entre tráfico normal y tráfico de ataque.

En el siguiente apartado se verá en detalle estos dos cambios realizados al algoritmo.

4.8 Algoritmo de centroides con tráfico particionado

Debido al problema del desbalance de datos, se realizaron dos cambios mayores en el algoritmo que se explican en los siguientes subapartados.

4.8.1 Número de centroides

El primero de ellos está relacionado con el número de centroides. Se decidió cambiar el número de centroides de tal forma que la cantidad de centroides de tráfico de ataque fuese muy superior a la de los demás tipos de tráfico. Por lo tanto, el número de agrupaciones en cada tipo de tráfico ya no sería el óptimo, diseñado mediante el uso del coeficiente de silueta, pero a cambio se tendría muchos más centroides de ataque en cada agrupación.

Este cambio en el número de centroides fue acompañado de una variación en el método de clasificar el tráfico. Tras hallar las distancias de todas las muestras a los distintos centroides, se calcula dentro de cada agrupación la distancia desde su centroide a la muestra más lejana, que será la distancia máxima dentro de la agrupación. En la clasificación, todas las observaciones serán clasificadas por defecto como tráfico normal a no ser que la distancia desde la observación a un centroide de ataque sea menor que la distancia máxima dentro de esa agrupación.

En ese caso la muestra será clasificada como ataque. Lo mismo ocurre para identificar las muestras de tráfico mixto pre-ataque y de tráfico mixto post-ataque.

Los números de centroides seleccionados en los distintos tipos de tráfico fueron los siguientes:

- Tráfico normal = 10 centroides
- Tráfico mixto pre-ataque = 30 centroides
- Tráfico mixto post-ataque = 30 centroides
- Tráfico de ataque = 80 centroides

Los números fueron elegidos arbitrariamente con las condiciones de que debía haber muchos menos centroides de tipo normal que de otro tipo, y que el número de centroides de ataque debería ser mucho mayor que los demás para así priorizar que las muestras de ataque se clasificasen correctamente.

4.8.2 Tráfico particionado

La segunda modificación consiste en alterar el algoritmo para que trate un segmento del tráfico cada vez que lo ejecutemos. En cada uno de los diez segmentos se analizan 4 situaciones según la posición del ataque en dicho segmento:

- Fin del ataque situado en el 5% de la duración del segmento
- Fin del ataque situado en el 30% de la duración del segmento
- Fin del ataque situado en el 60% de la duración del segmento
- Fin del ataque situado en el 90% de la duración del segmento

El motivo por el cual se escogieron estos valores fue para tener una muestra de cómo se comportaban los centroides si situábamos el ataque al principio, en el centro o al final del segmento.

4.8.3 Metodología

Primeramente, se trabaja clasificando los parámetros alfaestables de cada segmento de forma aislada para ver si era posible una clasificación dentro de cada uno. Se opera en cada segmento de forma similar a como se hizo para el tráfico completo mediante el algoritmo explicado en el apartado 4.5, pero esta vez sin detenernos a analizar los coeficientes de silueta e implementando los cambios en el método de la clasificación.

Esta clasificación fue realizada 4 veces para todos los segmentos: una vez con el ataque situado en la posición del 5% de la longitud del segmento, otra con el ataque situado en el 30% del segmento, otra en el 60% y otra en el 90%.

Las matrices de confusión obtenidas en todas las pruebas fueron satisfactorias ya que obtenía muy buenos resultados a la hora de clasificar todos los tipos de tráfico. A continuación, se muestra el mejor resultado y el peor de entre todas las matrices de confusión realizadas:

Matriz de confusion Alfa-Gamma-Delta quadratic							
True Class	ataque	1712			159	91.5%	8.5%
	mixtopost		287		11	96.3%	3.7%
	mixtopre			297	1	99.7%	0.3%
	normal		13	17	57721	99.9%	0.1%
		100.0%	95.7%	94.6%	99.7%		
		4.3%	5.4%	0.3%			
	ataque	mixtopost	mixtopre	normal	Predicted Class		

Figura 23: Mejor clasificación: Segmento n°4 con ataque situado en 30%.

Matriz de confusion Alfa-Gamma-Delta quadratic						
True Class	ataque	1587			284	<div><div>84.8%</div><div>15.2%</div></div>
	mixtopost		295		3	<div><div>99.0%</div><div>1.0%</div></div>
	mixtopre			292	6	<div><div>98.0%</div><div>2.0%</div></div>
	normal	85	918	260	56488	<div><div>97.8%</div><div>2.2%</div></div>
		94.9%	24.3%	52.9%	99.5%	
		5.1%	75.7%	47.1%	0.5%	
	ataque	mixtopost	mixtopre	normal		
	Predicted Class					

Figura 24: Peor clasificación: Segmento n°9 con ataque situado en 60%.

Como puede ver, en ambas clasificaciones se consigue unos porcentajes de acierto en el tráfico de ataque muy positivos, solamente es en el peor caso en el que se producen muchos falsos positivos para el tráfico mixto, confundiéndose con tráfico normal. Esto puede deberse a que el número de centroides de tráfico mixto no es suficientemente alto para este caso. Pero dado que en el resto de matrices de confusión el tráfico mixto se clasificaba relativamente bien, se eligieron estos números de centroides.

Tras haber observado que era posible una clasificación de los parámetros alfaestables segmento a segmento e independientemente de la posición del ataque, se analiza como extrapolar esta clasificación al tráfico completo. Para ello, se estudia la posición de los todos los centroides con los que se han realizado las clasificaciones anteriores para determinar si había alguna relación entre ellos que nos permitiese diferenciarlos según el tipo de tráfico al que pertenecen. Por tanto, se vuelven a realizar las ejecuciones del algoritmo, pero esta vez su único objetivo sería extraer los centroides generados por KM.

El objetivo de las nuevas ejecuciones del algoritmo era obtener los centroides de cada tipo de tráfico para un solo segmento. Por tanto, la metodología seguida para obtener todas las muestras de los centroides fue la siguiente:

- Primero, se sitúa el ataque en un punto en concreto del primer segmento (uno de los cuatro mencionados en el apartado 4.7.2), y se extraen sus centroides, que consistirán en 5 conjuntos de datos, uno para cada tipo de tráfico distinguiendo entre: tráfico normal antes del ataque, tráfico mixto pre-ataque, tráfico de ataque, tráfico mixto post-ataque y tráfico normal después del ataque.
- Esta misma operación será realizada para los siguientes segmentos del tráfico situando el ataque en el mismo punto del segmento, de forma que se ejecutará el algoritmo 10 veces en total (1 por segmento).
- Después de haber ejecutado el algoritmo 10 veces y haber obtenido 10 conjuntos de centroides por cada tipo de tráfico, se varía la posición del ataque dentro del segmento y se repite el proceso.
- Así, se ejecuta el algoritmo un total de 40 veces, dado que se evalúan 4 posiciones del ataque para cada uno de los 10 segmentos del tráfico.

El funcionamiento del nuevo algoritmo era similar al descrito anteriormente pero su objetivo era extraer los centroides de los segmentos en vez de clasificar sus parámetros alfaestables. Una vez obtenidos todos los centroides, fueron cargados y representados en otro script, el cual se utilizó para analizar las representaciones de los mismos y extraer resultados que se verán el siguiente capítulo.

4.9 Conclusiones

En este capítulo hemos revisado todos los procesos llevados a cabo durante el desarrollo del trabajo. Primero, comenzamos analizando los parámetros alfaestables de una trama de tráfico reducida y construimos un algoritmo para clasificarlos. Posteriormente comenzamos la segunda iteración del trabajo en la que analizamos la trama de tráfico completa, y realizamos los ajustes necesarios al algoritmo anterior para que trabajase correctamente con el nuevo conjunto de datos. Finalmente, extraemos los centroides con los que se han obtenido las clasificaciones para analizarlos y sacar las conclusiones que se exponen en el siguiente capítulo.

5 Pruebas y resultados

5.1 Introducción

En este capítulo se presentan los resultados de los centroides obtenidos mediante el uso del algoritmo explicado en el capítulo anterior y se explica la manera en la que han sido analizados.

5.2 Metodología

Tras obtener todos los resultados de las ejecuciones del algoritmo se procede a representarlos. Para analizar los resultados y encontrar similitudes entre ellos, los centroides serán analizados de dos formas: según la posición del ataque y según el segmento.

- Según la posición del ataque: Se generan 4 gráficas (una por cada posición del ataque) y en cada una se representan los centroides de ataque de todos los segmentos para una posición del ataque determinada. El objetivo de estas representaciones fue observar si el cambio en la posición del ataque tenía una influencia notable en la posición de los centroides de ataque.
- Según el segmento: Se generan 10 gráficas (una por cada segmento del tráfico) y en cada una de ellas se representan los centroides de cada segmento para todas las posiciones del ataque. En cada gráfica se muestran los centroides de ataque y los centroides de normal con el objetivo de ver si hay zonas en las que estos dos tipos de centroides sean claramente diferenciables.

Después de observar las numerosas representaciones, analizar las zonas en la que se sitúan los centroides de ataque y ver cómo se comportan estos según los dos criterios expuestos anteriormente, es necesario construir ahora una simulación en la que el tráfico sea analizado de forma completa y el ataque sólo se produzca un punto de toda la trama para comprobar que los resultados y las conclusiones obtenidas se confirmen en el primer escenario de estudio (una trama de tráfico real con un sólo ataque).

Como se realizó al comienzo del trabajo, situaremos el ataque en un sólo punto del tráfico completo y extraeremos los centroides de los segmentos del tráfico para ver si de esta forma cumplen las características observadas en las representaciones anteriores. Esta vez, 9 de los 10 segmentos estarán libres de ataque y sólo uno contendrá centroides de ataque y de tipo mixto.

5.3 Resultados

Las gráficas generadas son tridimensionales ya que los centroides vienen definidos por los tres parámetros alfaestables que hemos estudiado a lo largo del trabajo. (alfa, gamma y delta).

A continuación, se muestran las gráficas con los centroides de ataque según los dos criterios vistos:

Según la posición del ataque:

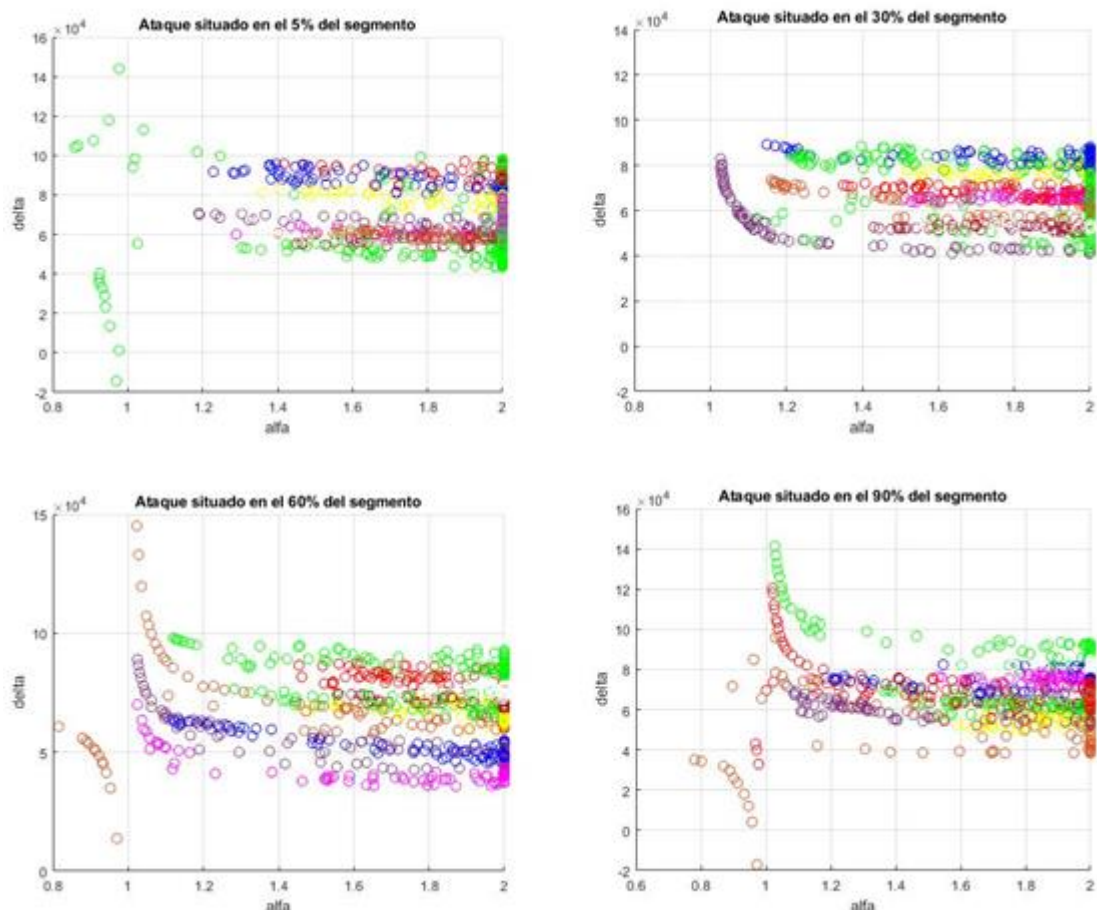


Figura 25: Centroides de ataque según la posición del ataque.

Las gráficas adjuntadas solamente muestran el plano alfa-delta dado que fue el que aportaba más información en un principio. Cada color se corresponde con 1 de los 10 segmentos del tráfico particionado. Se puede observar que las muestras de cada segmento se distribuyen de forma horizontal en las gráficas mostradas, formando conjuntos de muestras de distintos colores. Esto se aprecia mejor en la gráfica del 60% y ocurre debido a que la variación relativa de las muestras en alfa es mucho mayor a la variación en delta. Esta observación podría haber resultado útil pero dado que los conjuntos de muestras no tienen un orden particular, sino que cambia según la representación no conseguimos sacar ninguna conclusión de ellas.

Una curiosidad que salto a la vista fue la asíntota que se produce en el valor de alfa igual a 1, pero tampoco resultó de utilidad para sacar conclusiones sobre el comportamiento de los centroides.

Según el segmento

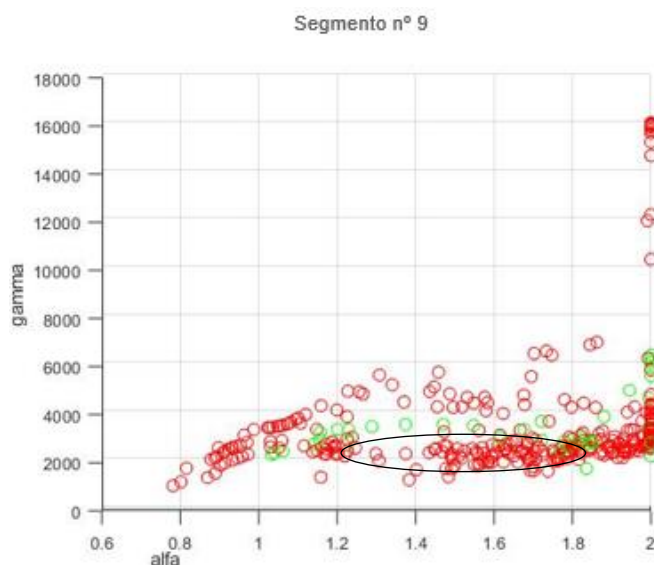


Figura 26: Centroides del segmento nº9.

En la gráficas anterior se muestran los centroides de ataque (en rojo) y los centroides de tipo normal (en verde) para el segmento número 9 del tráfico. La representación muestra el plano alfa-gamma, ya que en este plano se podían distinguir mejor las zonas de ataque de las zonas de tráfico normal. Podemos observar como la mayor parte de los valores de gamma de los centroides de ataque se comprenden entre 2000 y 3000.

Adicionalmente, para dichos valores de gamma y los valores de alfa 1.8 y 1.2 encontramos una zona en el plano en la que sólo se observan centroides de ataque, quedando los centroides de ataque excluidos de la zona.

Esta zona exclusiva de centroides de ataque se puede observar en el plano alfa-gamma de las representaciones de los centroides de todos los segmentos, por lo que dicha zona es independiente del segmento.

El siguiente paso fue realizar una prueba situando el ataque en un solo punto del tráfico completo para comprobar que esta zona de centroides de ataque también se aprecia en el escenario de estudio original. El ataque fue situado en el centro del tráfico, es decir, en el quinto segmento, dejando todos los demás libres de tráfico de ataque. En las siguientes gráficas observaremos una vista previa de la gráfica tridimensional alfa- delta-gamma y se mostrarán el plano alfa-gamma que contienen los centroides de ataque y de tipo normal extraídos del tráfico completo.

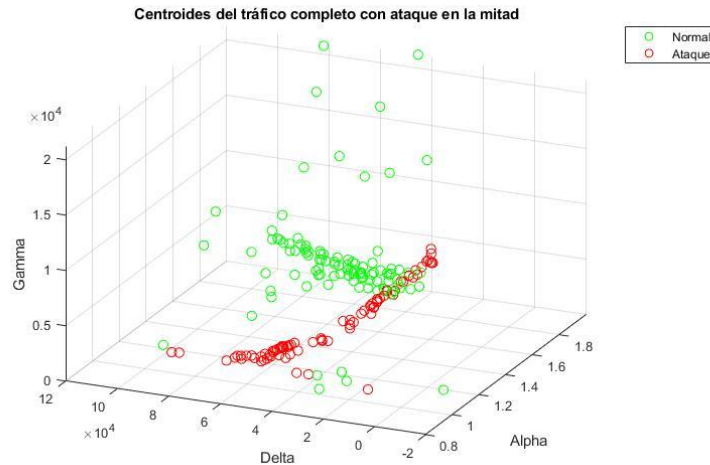


Figura 27: Representación 3D de la comprobación

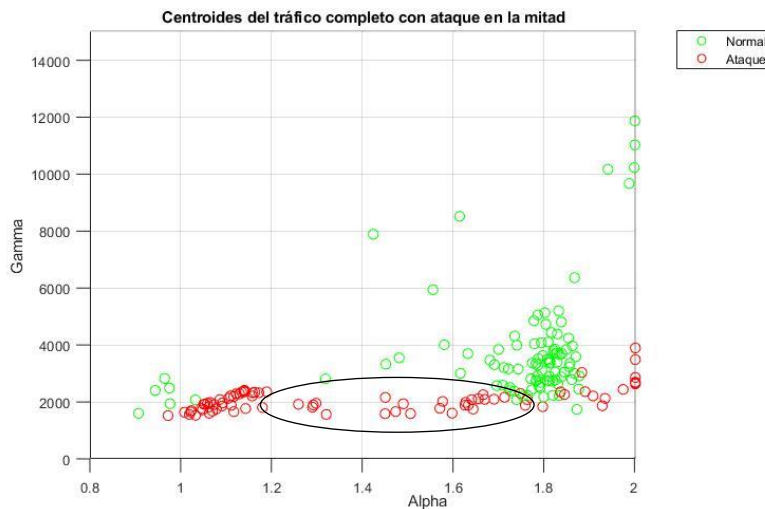


Figura 28: Representación de Alfa frente a Gamma de la comprobación

En las representaciones estamos observando 80 centroides de ataque y 100 de tipo normal, ya que en cada segmento sacamos 10 centroides de este tipo. A pesar del aumento en el número de centroides de tipo normal, en la gráfica de alfa-gamma podemos ver que la zona de la que previamente hemos hablado (entre los valores de gamma 2000 a 3000 y entre los valores de alfa 1.2 a 1.8) también está libre de centroides de tipo normal como habíamos visto. Por lo tanto, hemos encontrado unos valores de gamma y de alfa entre los cuales solo existirán centroides de ataque, independientemente de cuando se produzca el ataque.

5.4 Conclusiones

En este capítulo hemos presentado gráficamente los resultados obtenidos tras analizar de dos formas distintas los centroides extraídos de todos los segmentos del tráfico y hemos dado con una zona en la cual sólo recaen centroides de ataque, quedando dicha zona libre de centroides de tipo normal.

6 Conclusiones y trabajo futuro

6.1 Conclusiones

En este trabajo se ha partido de dos estudios anteriores sobre los parámetros alfaestables de un ataque sintético de tipo DDoS en una trama real de tráfico. En el TFG anterior se realizó una segmentación de los parámetros alfaestables diferenciando tráfico normal y tráfico de ataque.

Continuando con el trabajo realizado anteriormente, se volvió a realizar dicha segmentación de los parámetros, pero esta vez distinguiendo entre más tipos de tráfico: tráfico normal, tráfico mixto y tráfico de ataque. Esta primera iteración del trabajo sirvió para excluir beta de los parámetros alfaestables sujetos a estudio ya que no resultaba útil para nuestro algoritmo de CL, y para ver que era posible realizar una clasificación efectiva en una trama reducida del tráfico utilizando alfa, delta y gamma. El descarte de beta fue acertado puesto que al trabajar con el tráfico completo se vio que beta no siempre tomaba el valor de 1 para muestras de ataque, por lo tanto, no resultaba útil tampoco a la hora de clasificar el tráfico.

En la segunda iteración del trabajo se estudio la trama de tráfico completa mediante los otros tres parámetros alfaestables, haciendo modificaciones a los algoritmos anteriores para solucionar el problema del desbalance de datos que se dio al tener una cantidad de tráfico normal mucho mayor a la de ataque.

Estos cambios también implicaron una partición del tráfico completo en 10 segmentos de tráfico en los cuales se estudiaron el comportamiento de los centroides extraídos por KM. Se analizó el efecto que tenía sobre los centroides de ataque el cambio de la posición del ataque, pero no se pudo identificar ningún patrón que nos permitiese sacar conclusiones al respecto.

También se estudió en cada segmento como se distribuyen los centroides de ataque frente a los centroides de tráfico normal, y se encontró una zona en el plano alfa-gamma, concretamente entre los valores de gamma de 2000 a 3000 y los valores de alfa de 1.2 a 1.8, en la que sólo recaen centroides de ataque, quedando excluidos los centroides de tipo normal para todos los segmentos analizados.

Esta zona libre de centroides de tipo normal podría facilitar la creación de un detector de ataques el cual identificase si los parámetros alfaestables del tráfico recaen sobre la zona en cuestión o fuera de ella. Quizás no fuese un método perfecto para clasificar el tráfico, pero sí podría ser útil para identificar un ataque, puesto que es muy posible que algunos de los parámetros alfaestables de ataque estén aislados en dicha zona del plano alfa-gamma.

6.2 Trabajo futuro

Este trabajo abre nuevos horizontes dentro de esta línea de investigación. El trabajo futuro que se podría desarrollar a partir del actual podría tratar los siguientes temas:

- Generar una predicción del tráfico mediante el método Holt-Winters.
- Revisar la distribución de los valores de los parámetros alfaestables del tráfico.
- Utilizar la función generadora de momentos para la distribución alfaestable.

Referencias

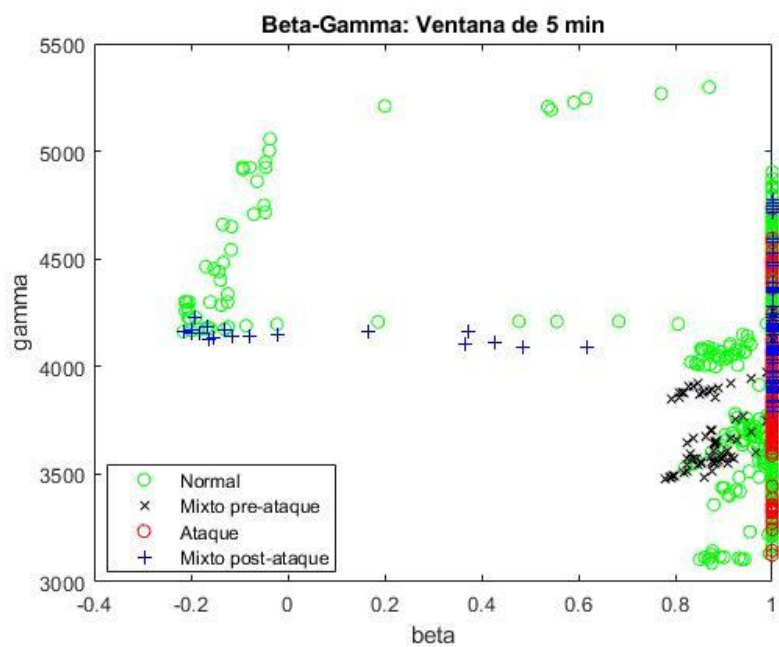
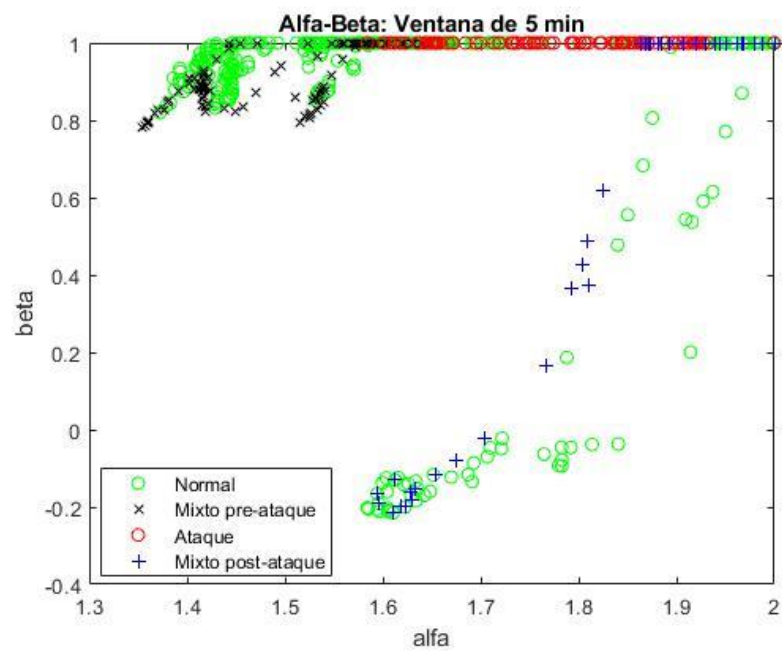
- [1] <https://es.wikipedia.org/wiki/Ciberataque>. Consultado en mayo 2020
- [2] <https://www.osi.es/es/actualidad/blog/2018/08/21/que-son-los-ataques-dos-y-ddos>. Consultado mayo 2020
- [3] https://es.wikipedia.org/wiki/Distribuci%C3%B3n_estable. Consultado en mayo 2020
- [4] <https://es.wikipedia.org/wiki/Algoritmo>. Consultado en mayo 2020
- [5] Paloma Recuero de los Santos. “Tipo de aprendizaje en Machine Learning: supervisado y no supervisado”. Noviembre 2017.
- [6] Ligdi González. “Diferencia entre algoritmos de clasificación y regresión”. Junio 2018
- [7] Victor Román. “Aprendizaje Supervisado: Introducción a la Clasificación y Principales Algoritmos”. Marzo 2019.
- [8] Fernando Sancho Caparrini. “Algoritmos de Clustering”. Diciembre 2019.
- [9] Santiago Bancho. “Calidad del agrupamiento: Coeficiente de silueta”. Universidad Nacional de Lujan. 2015.
- [10] Alberto Ruiz Santos. “Segmentación de tráfico en Internet mediante la clasificación de parámetros estadísticos”. Trabajo Fin de Grado. Julio 2019.
- [11] Clusterdata (). Función de la librería de “*Statistics and Machine Learning Toolbox*” de MATLAB.
- [12] Kmeans(). Función de la librería de “*Statistics and Machine Learning Toolbox*” de MATLAB.

Glosario

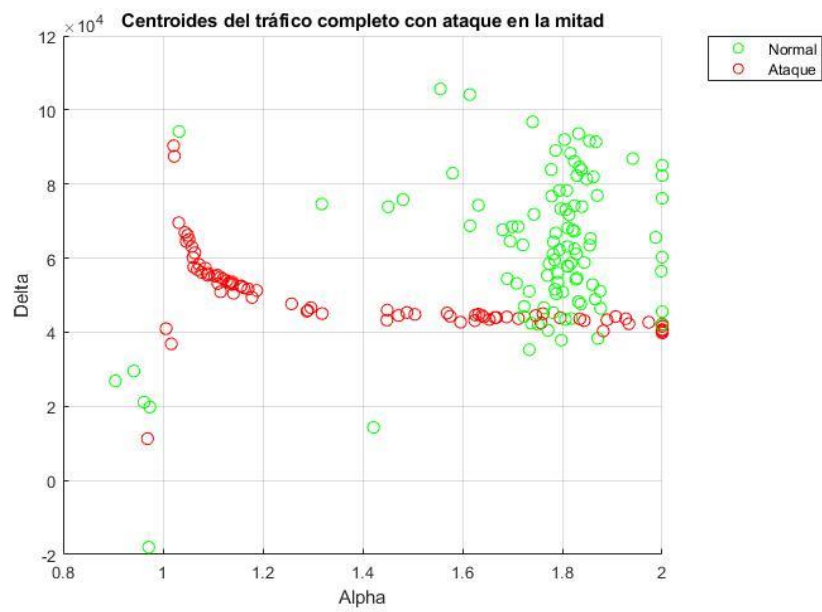
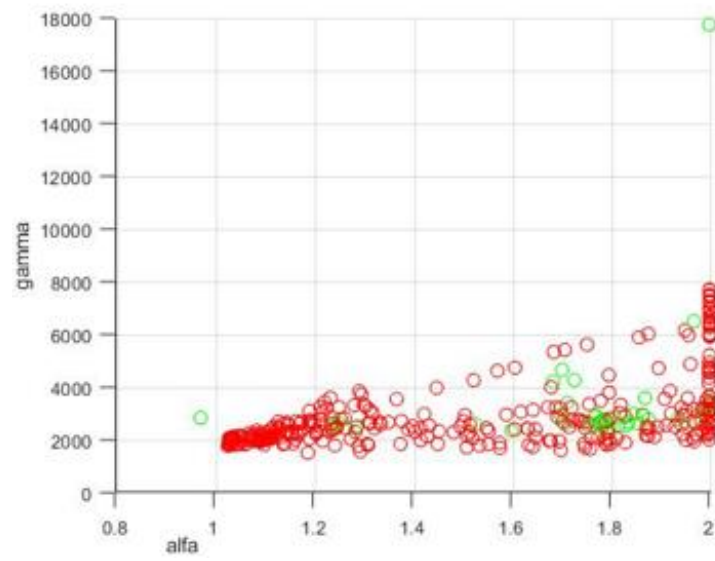
DoS	Denial of Service
DDoS	Distributed Denial of Service
ML	Machine Learning
CL	Clustering
KM	K-Means

Anexos

A Gráficos adicionales



Segmento nº 8



B Repositorio del código generado

Todo el código generado se ha puesto público en un repositorio para que pueda ser consultado en la siguiente dirección: <https://github.com/erevuelta/TFG>